

**Università degli Studi di Napoli  
Federico II**

**Contributions to Symbolic Data Analysis:  
*A Model Data Approach***

**Simona Signoriello**

Tesi di Dottorato di Ricerca in  
Matematica per l'analisi economica e la finanza

*XXI Ciclo*





Contributions to Symbolic Data Analysis  
*A Model Data Approach*

Napoli, 1 Dicembre 2008



# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Symbolic Data</b>	<b>5</b>
1.1 Interval Data . . . . .	8
1.2 Histogram Data . . . . .	11
1.3 A new type of symbolic data: “Model Data” . . . . .	15
<b>2 A review of Symbolic Data Analysis</b>	<b>19</b>
2.1 Principal Components Analysis for Symbolic Data . . .	19
2.1.1 Principal Components Analysis of Symbolic Data described by intervals . . . . .	20
2.1.2 Generalization of the Principal Components Analy- sis to Histogram Data . . . . .	22
2.2 Clustering of Symbolic Data . . . . .	25
2.2.1 Dissimilarity measures of Interval Data . . . . .	26
2.2.2 Distance measures between histogram data . . .	32
<b>3 Model Data Building</b>	<b>35</b>
3.1 Mathematical Model . . . . .	36
3.2 Basis functions approach . . . . .	38
3.2.1 Polynomial basis functions . . . . .	39
3.2.2 Piecewise Polynomial Basis . . . . .	40

3.2.3	Spline functions . . . . .	41
3.2.4	B-spline . . . . .	42
3.3	How to approximate a histogram using a B-spline . . .	45
3.4	Histogram transformation process . . . . .	49
<b>4</b>	<b>Model Data Analysis</b>	<b>55</b>
4.1	Multiple Factor Analysis . . . . .	55
4.2	Clustering analysis for Model Data . . . . .	59
4.2.1	The Inter-Model Distance . . . . .	59
4.2.2	Clustering utility functions . . . . .	62
4.2.3	Distance between Model Data . . . . .	64
<b>5</b>	<b>A case study on real Data</b>	<b>67</b>
5.1	Data Structure . . . . .	68
5.2	Case study: Multiple Factor Analysis . . . . .	70
5.3	Case study: Cluster analysis . . . . .	80
	<b>Conclusioni</b>	<b>89</b>
<b>A</b>	<b>Routine in Matlab Language</b>	<b>95</b>
A.1	Model Data Building . . . . .	96
A.2	Cluster Analysis on Model Data . . . . .	102
	<b>Bibliography</b>	<b>107</b>

# Introduction

This work aims at analysing complex phenomena through the construction of appropriate models, followed by an analysis of the characteristic parameters of the models. It all derives from the need to work, not only with empirical values but with functions that are able to smooth the histogram and give us the possibility to omit values that could be outlier. According to the classical theory of measure, the data generated by a “correct” model are more “real” than the empirical one, because they are purified from error sampling and from error of measurement.

We should never forget that there are no “real” models, but rather models that approximate the reality in a more or less accuracy.

Models compatible with empirical data can be manifold.

The idea proposed in this thesis is to transform the histogram data by means of an approximation function in order to control the error deriving from empirical data. According to the paradigm:

$$DATA = MODEL + ERROR$$

we may actually think to work on the model and keep under control the error term at the same time.

What we look for is the right compromise between model and error. Our target is to be able to work with models that are comparable in order to be able to apply the techniques of a Multidimensional Data

Analysis. For that reason, all the histograms will be transformed into models through the approximation by means of functions of the same family. In that case we would work with data that have been synthesized through a model, and from there we would obtain  $N$  models for each variable, all corresponding to the  $i$ -th observation. Models constructed that way can be synthesized through parameters and through an appropriate quality index of adaptation. Successively we will pass on to the analysis of the data achieved through adequate techniques of Multidimensional Analysis.

This thesis has been divided into five chapters.

In the *first chapter* I will present a concise introduction to symbolic data, with particular attention to interval data and histogram data. Then I will introduce the subject matter of the thesis, considered to be a new type of symbolic data called “Model Data”.

In the *second chapter* I will present a review of the methods of symbolic Analysis for interval data and for histogram data, already introduced, with particular attention paid to the techniques of Principal Components Analysis and to Cluster Analysis.

The *third chapter* is the fulcrum of the thesis and here the construction of “Model Data” is demonstrated. I will furthermore introduce some basic definitions of interpolation models and approximates with a short review of the most common base functions, polynomial functions, piecewise polynomial functions, spline functions and B-spline functions, which then will be used for the approximation of the histograms, arriving in the end to form the parameters that set up the “Model Data”.

In the *fourth chapter* I will show how to analyze these data and how to carry out a Principal Components Analysis (PCA) with subse-



quent cluster. The proposed technique is a Multiple Factor Analysis meant to substitute the classical PCA, since we work with block matrix where each of them is created by the parameters of the models for each variable. On the other hand to carry out a Cluster Analysis, I have used an interval between models already proposed earlier by Lauro, Romano, and Giordano in 2006, but re-adapted for the parameters available.

A case study is presented in the *fifth chapter* based on real data. Particular financial data have been used, that is, a database, referring to 30 stocks of the S&P MIB, which has registered some variables from 2004 to 2005 like closing prices, opening prices, daily minimum and maximum prices, adjusted closing prices, and volumes. All the methodologies proposed through the routines constructed in Matlab and the use of the X1-stat. packet have been applied on this database.



# Chapter 1

## Symbolic Data

In real life, the use of single valued variables could lead to a loss of information. For example, daily temperatures registered as the variation between the minimum and the maximum values should provide a more realistic view of the weather conditions than the daily average value. Many application fields take advantage of the statistical analysis of the interval data, such as the weather condition analysis, statistical quality control, financial data analysis, etc. Due to recent developments in data warehousing, a huge amount of continuous data are stored at any occurrence (such as: Stock Exchange Data).

Traditionally, real-valued vectors have been used to model participants of a specific domain. If  $n$  individuals are evaluated by  $m$  variables, then a  $n \times m$  matrix will hold all the relationships between them. However, the real world is too complex to be described in this relatively simple tabular model. In order to deal with more complex cases we use symbolic data. In this context, data are not confined to be real values, but can be selected from a wider list: set-, interval-, histogram-, tree-, graph, function, fuzzy data, etc. Classical data on  $p$  random variables are represented by a single point in  $p$ -dimensional space  $\mathfrak{R}^p$ . In contrast, symbolic data with measurements on  $p$  random

variables are  $p$ -dimensional hypercubes (or hyperrectangles) in  $\Re^p$ , or a Cartesian product of  $p$  distributions, broadly defined. The “hypercube” would be a familiar four-sided rectangle if, for example,  $p = 2$  and the two random variables take values over an interval, say,  $[a_1, b_1]$  and  $[a_2, b_2]$ , respectively. In this case, the observed data value is the rectangle  $R = [a_1, b_1] \times [a_2, b_2]$  with vertices  $(a_1, a_2)$ ,  $(a_2, b_2)$ ,  $(b_1, a_2)$  and  $(b_1, b_2)$ . However, the  $p = 2$  dimensional hypercube need not be a rectangle; it is simply a space in the plane. A classical value as a single point is a special case. Instead of an interval, observations can take values that are lists, e.g., {good, fair} with one or more different values in the list. Or, the observation can be a histogram. Indeed, there are many possible formats for symbolic data.

Basic descriptions of symbolic data such as interval data and histogram data are covered in this chapter, before going on to specific analytic methodologies in the chapter that follow. At the outset, however, it is observed that a symbolic observation in general has an internal variation. For example, an individual whose observed value of a random variable is  $[a, b]$ ,  $a \neq b$ , is interpreted as taking (several) values across that interval. This is not to be confused with uncertainty or impression when the variable takes a (single) value in that interval with some level of uncertainty. A classical observation with its single point value perforce has no internal variation, and so analyses deal with variation between observations only. In contrast, symbolic data deal with the internal variation of each observation plus the variation between observations.

Symbolic data arise in a variety of different ways. Some data are inherently symbolic. For example, it may not be possible to give the exact cost of an apple (or shirt, or product, or ...) but it is only its cost that takes values in the range  $[16, 24]$  cents (say). We also note that an interval cost of  $[16, 24]$  differs from that of  $[18, 22]$  even though these two intervals both have the same midpoint value of 20. A classical analysis using the same midpoint (20) would lose the fact

---

that these are two differently valued realizations with different internal variations.

In another direction, an insurance company may have a database of hundreds (or millions) of entries each relating to one transaction for an individual, with each entry recording a variety of demographic, family history, medical measurements, and the like. However, the insurer may not be interested in any one entry per se but rather is interested in a given individual (Colin, say). In this case, all those single entries relating to Colin are aggregated to produce the collective data values for Colin. The new database relating to Colin, Mary, etc., will perforce contain symbolic-valued observations. For example, it is extremely unlikely that Mary always weighed 125 pounds, but rather that her weight took values over the interval  $[123, 129]$ , say.

In these two types of settings, the original database can be small or large. A third setting is when the original database is large, very large, such as can be generated by contemporary computers. Yet these same computers may not have the capacity to execute even reasonably elementary statistical analyses. For example, a computer requires more memory to invert a matrix than is needed to store that matrix. In these cases, aggregation of some kind is necessary even if only to reduce the dataset to a more manageable size for subsequent analysis. There are innumerable ways to aggregate such datasets. Clearly, it makes sense to seek answers to reasonable scientific questions and to aggregate accordingly. As before, any such aggregation will perforce produce a dataset of symbolic values. In the following sections we focus attention on two type of symbolic data for quantitative variables: the interval data and the histogram data that is the case in which we do not have any information about the internal variation interval and the case in which we have an additional information coming from the aggregation that leads us to construct a histogram.

## 1.1 Interval Data

A generic interval variable  $Y$  represents a set of bounded intervals:

$$Y_j = [\underline{y}_j, \overline{y}_j], j = 1, \dots, n$$

where  $\underline{y}_j$  and  $\overline{y}_j$  represent the lower (min) and the upper (max).

To operate on intervals, the algebra defined by Moore (1957) is used.

It defines the basic arithmetic operations in the following way:

If  $\bullet$  is one of the symbols  $+$ ,  $-$ ,  $\times$ ,  $\div$ , we define arithmetic operations on intervals by:

$$[a, b] \bullet [c, d] = \{x \bullet y : a \leq x \leq b, c \leq y \leq d\} \quad (1.1)$$

It is not defined the division  $[a, b] \div [c, d]$  when  $0 \in [c, d]$ .

The sum, the difference, the product, and the ratio (when defined) between two intervals is the set of the *sums*, the *differences*, the *products*, and the *ratios* between any two elements from the first and the second interval, respectively.

An *equivalent definition*: let  $\mathfrak{S}$  be the set of intervals, and let  $[a, b], [c, d]$  be elements of  $\mathfrak{S}$ , it is:

- $[a, b] + [c, d] = [a + c, b + d]$
- $[a, b] - [c, d] = [a - d, b - c]$
- $[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$
- if  $0 \notin [c, d]$ , then  $[a, b] \div [c, d] = [a, b] \times [1/d, 1/c]$

Because of the complexity of multidimensional interval data some specific tools are needed:

- intervals coding into single valued data (box vertices, midpoints and radii) and ex-post intervals reconstruction in in order to visualize the output;

- intervals direct treatment with a suitable algebra and/or algorithms.

Many authors have dealt with interval data analysis by the encoding in midpoint (Midpoint & Radii PCA [48]), radii or vertices (Vertices Principal Components Analysis (V-PCA) [11]); Symbolic Objects PCA, [42]) Some authors ([50], [33]) have defined descriptive statistics for interval variables in analogy to the case of single-valued data. In particular, in [33] there are described interval statistics, such as mean and deviation from mean and it is showed that they share the same properties of the corresponding statistics for single-valued data. Therefore the basic idea is to rewrite statistics for interval data in the same way as for single-valued data.

Unfortunately, the interval algebra was born in the field of error-theory where intervals are very small, but this is not longer true for Statistical Interval.

First of all the so-called wrapping effect leads to wider intervals than they actually should be. This effect induces a distinction between “interval of solutions” and the “interval solutions”.

To clarify the wrapping effect we consider the following example:

Let the function  $f(x) = x(x - 1)$  be considered with  $0 < x < 1$ .

By means of algebra intervals we have the following results:

$$f([0, 1]) = [0, 1]([0, 1] - 1) = [0, 1] - [-1, 0] = [-1, 0]$$

By calculating instead, by means of the classical analysis, the real range of variation of the function  $f$ , that is the set:

$$f([0, 1]) = \{f(x)/x \in [0, 1]\}, \text{ it gives the interval } [-1/4, 0] \subset [-1, 0].$$

Therefore, we can observe that the result reached with the algebra of intervals is actually a broader range containing the exact range of variation of the function here considered.

However, under some conditions, it is possible to obtain the same range of variation:

**Proposition.** *If  $f(x_1, \dots, x_n)$  is a real rational function in which each variable  $x_i$  occurs only once and only at the first power, then the corresponding interval expression  $f(X_1, \dots, X_n)$  will compute the actual range of the values of  $f()$  for  $x_i$  in  $X_i$ :*

$$f(X_1, \dots, X_n) = \{y/y = f(x_1, \dots, x_n), x_i \in X_i, i = 1, \dots, n\}.$$

For example: Let  $f(x) = x/(x - 2)$  be a real function, then:

$$f([10, 12]) = \frac{[10, 12]}{[10, 12] - 2} = \frac{[10, 12]}{[8, 10]} = [1, 1.5]$$

But the actual range is  $[1.2, 1.25] \in [1, 1.5]$ .

Making the transformation:

$$f(x) = \frac{x - 2}{x - 2} + \frac{2}{x - 2} = 1 + \frac{2}{x - 2}$$

and calculating in  $[10, 12]$  it is obtained:

$$f([10, 12]) = 1 + \frac{2}{[10, 12] - 2} = 1 + \frac{2}{[8, 10]} = 1 + [0.2, 0.25] = [1.2, 1.25].$$

In general, it is not always possible to write a rational expression, in which a number of real variables is larger than one, so that the new expression contains a single occurrence of each variable. We say that, in a way, we are obliged to have an interval that contains the effective range of the function.

Another issue that was born with interval algebra is the fact that, by the arithmetic point of view, it is not possible to make all the operations similar to the case of single-valued data, in the sense that there is no-correspondence between the two cases, because many arithmetic



properties do not hold anymore.

For example, let consider the sum of two intervals:

$$[1, 2] + [3, 4] = [4, 6]$$

According to the properties of the classical arithmetic, subtracting the second addend from the sum we get the first one, but in this case this does not occur:

$$[4, 6] - [3, 4] = [0, 3] \neq [1, 2]$$

Therefore, it is impossible to adapte all statistic formulations that apply to the single-valued data to interval-data without making the appropriate amendments.

As said before, it is clear that speaking of variance function for interval-valued data some complications arise, because we are dealing with a quadratic function. Therefore, several authors have treated the calculation of the variance through numerical algorithms [33] and by optimization techniques [31].

Moreover, in the interval data it is supposed that all the values within the interval have the same probability, in other words, it is supposed to have a uniform distribution. But in real cases we can have a different probability distribution within the interval, so it is possible to work with histogram data that give us additional information about the variance within the interval.

## 1.2 Histogram Data

In many real experiences, data are collected and/or represented by frequency distributions. If  $Y$  is a numerical and continuous variable, many distinct values  $y_i$  can be observed. In these cases, the values are usually grouped in a smaller number  $H$  of consecutive and disjoint bins  $I_h$  (groups, classes, intervals, etc.). The frequency distribution of

the variable  $Y$  is given considering the number of data values  $n_h$  falling in each  $I_h$ . The histogram is then the typical graphical representation of the variable  $Y$ . The interest to analyze data expressed by frequency distributions as well as by histograms, is apparent in many fields of research. In particular, we may refer to the treatment of experimental data that are collected in a range of values, whereas the measurement instrument gives only approximated (or rounded) values. An example can be given by sensors for air pollution control located in different zones of an urban area. The different distributions of measured data about the different levels of air pollutants during a day, allow us to compare, and then to group them into homogeneous clusters, the different controlled zones.

In a different context of analysis, histograms are the key to understanding digital images. A digital image is basically a mosaic of square tiles or “pixels” of uniform color that are so tiny that the composed image appears uniform and smooth. Instead of sorting them by colour, they can be sorted into 256 levels of brightness from black (value 0) to white (value 255) with 254 gray levels in between. The height of each vertical “bar” tells you how many pixels there are for that particular brightness level.

Let  $Y$  be a continuous variable defined on a finite support  $S = [\underline{z}; \bar{z}]$ , where  $\underline{z}$  and  $\bar{z}$  are the minimum and maximum values of the domain of  $Y$ . The variable  $Y$  is supposed partitioned into a set of contiguous intervals (bins)  $I_1, \dots, I_h, \dots, I_H$ , where  $I_h = [\underline{z}_h; \bar{z}_h)$ . Given  $N$  observations on the variable  $Y$ , each semi-open interval,  $I_h$ , is associated with a random variable equal to  $\phi(I_h) = \sum_{u=1}^N \phi_{z_u}(I_h)$  where  $\phi_{z_u}(I_h) = 1$  if  $z_u \in I_h$  and 0 otherwise. Thus, it is possible to associate to  $I_h$  an empirical distribution  $\pi_h = \phi(I_h)/N$ . A histogram of  $Y$  is then the graphical representation in which each pair  $(I_h; \pi_h)$  (for  $h = 1, \dots, H$ ) is represented by a vertical bar, with base interval  $I_h$  along the horizontal axis and the area proportional to  $\pi_h$ . Consider  $E$  as a set of  $n$  empirical distributions  $Y(i)$  ( $i = 1, \dots, n$ ).

## 1.2. Histogram Data

---

Specifically, for a generic variable, the  $i$ -th histogram data is a model to represent an empirical distribution described as a set of  $H$  ordered pairs  $Y(i) = (I_h, \pi_h)$  as:

$$I_{hi} \equiv [\underline{z}_{hi}, \bar{z}_{hi}] \quad \underline{z}_{hi} \leq \bar{z}_{hi} \in \mathfrak{R},$$

$$\bigcup_{h=1, \dots, H} I_{hi} = [\min_{h=1, \dots, H} \{\underline{z}_{hi}\}, \max_{h=1, \dots, H} \{\bar{z}_{hi}\}],$$

$$\pi_h \geq 0,$$

$$\sum_{h=1, \dots, H} \pi_h = 1.$$

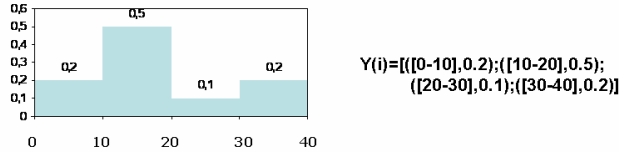


Figure 1.1: Example of histogram data and its representation

This kind of data, compared to the interval data enjoy a two-dimensional representation, where horizontally the subdivision in intervals is represented and vertically there are the respective densities. In particular, the histogram data (see figure 1.1) can be seen as symbolic data divided into many intervals. On each of them it is specified additional information about its relative frequency (or density of frequency). Therefore, we might consider working with intervals of different weight given by the respective frequency and, from this, build proper descriptive statistics and appropriate distances between intervals, as long as we choose intervals small enough to be able to make use of the theory of interval algebra, as specified in the previous

section. In particular, in literature a histogram arithmetic was proposed by Colombo and Jaarsma (1980):

Given two histograms,  $Y_A = (I_{Ah}, \pi_{Ah})$  with  $h = 1, \dots, n$  and  $Y_B = (I_{Bh'}, \pi_{Bh'})$  with  $h' = 1, \dots, m$  both representing a pairs of independent random variables  $A$  and  $B$ , and  $\bullet$  being some arithmetic operator in  $\{+, -, \times, \div\}$ ,  $C = A \bullet B$  can be approximated by the unsorted histogram  $Y_C = (I_{Ck}, \pi_{Ck})$  with  $k = 1, \dots, n \cdot m$ , where

$$\underline{z}_{C(h-1)m+h'} = \min \{ \underline{z}_{Ah} \bullet \underline{z}_{Bh'}, \bar{z}_{Ah} \bullet \underline{z}_{Bh'}, \underline{z}_{Ah} \bullet \bar{z}_{Bh'}, \bar{z}_{Ah} \bullet \bar{z}_{Bh'} \},$$

$$\bar{z}_{C(h-1)m+h'} = \max \{ \underline{z}_{Ah} \bullet \underline{z}_{Bh'}, \bar{z}_{Ah} \bullet \underline{z}_{Bh'}, \underline{z}_{Ah} \bullet \bar{z}_{Bh'}, \bar{z}_{Ah} \bullet \bar{z}_{Bh'} \}, \text{ and}$$

$$\pi_{C(h-1)m+h'} = \pi_{Ah} \bullet \pi_{Bh'}$$

Also in this case there are some disadvantage:

- some of the constituent intervals of the resultant may overlap;
- making a series of arithmetic operations on a number of histograms, the resulting histogram is expected to have a high number of intervals;
- condensing the unsorted histograms obtained after each operation into histograms of  $l \ll n \cdot m$  intervals in order to avoid an enormous final number of intervals;
- histogram arithmetic subsumes interval arithmetic, which in turn, subsumes classic arithmetic.

Moreover, the disadvantage to use the histogram data is that we have an empirical distribution, so our propose is to model the histogram through a suitable function that represent the shape of the distribution purified from the error.

In the next section you will be introduced to the new proposed data that we have called “Model Data”.

## 1.3 A new type of symbolic data: “Model Data”

According to the classical theory of measure, the data generated by a “correct” model are more “real” than the empirical one, because they are purified from error sampling and from error of measurement. We should never forget that there are no “real” models, but rather models that approximate the reality in a more or less accurate manner.

Models compatible with empirical data can be manifold.

The idea proposed in this thesis is to transform the histogram data by means of an approximation function in order to control the error deriving from empirical data.

According to the paradigm:

$$DATA = MODEL + ERROR$$

we may actually consider working on the model and keep the error term under control at the same time.

Therefore, provided that our data have been suitably processed as a function, they may be summarised through the function parameters and some indices of goodness of fit.

In this case, we will work on data summarized through a function, so for each variable you will get  $N$  functions, each of which corresponds to the  $i$ -th observation. Schematically it can be summarised as in the figure 1.2. The new data have to be proportional in number to the parameters of the function, in this way any function will be replaced by its own parameters and our data will be as many as the  $units \times variables \times number\ of\ parameters$ . Assuming that all the functions have  $k$  parameters:  $b_1, \dots, b_k$  and an appropriate index ( $I$ ) of goodness of fit, we can summarise the data as in the figure 1.3.

The problem is now to derive some functions that, from a mathe-

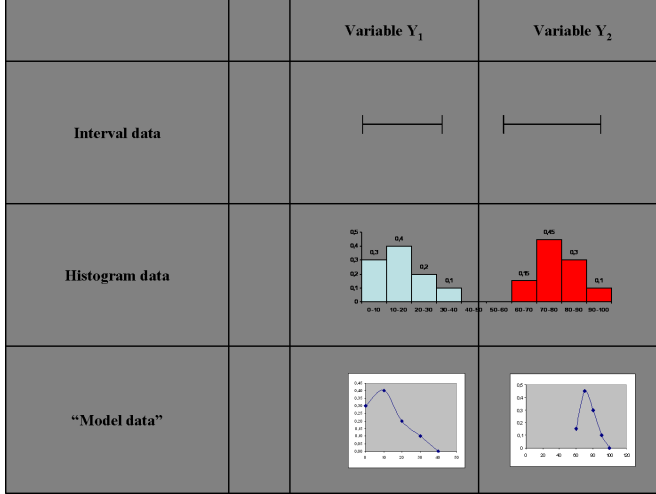


Figure 1.2: Different type of Symbolic Data

mathematical point of view, are the best approximations of the data, such as the spline or B-spline; therefore we need to identify which one could have the best grade of the interpolator function.

How "Model Data" are built will be presented in the third chapter.

### 1.3. A new type of symbolic data: “Model Data”

---

	Parameter Estimates		
	Var1	Var2	Var3
Osservazione1	$b_{111}, b_{112}, \dots, b_{11k}, I_{11}$	$b_{121}, b_{122}, \dots, b_{12k}, I_{12}$	$b_{131}, b_{132}, \dots, b_{13k}, I_{13}$
Osservazione2	$b_{211}, b_{212}, \dots, b_{21k}, I_{21}$	$b_{221}, b_{222}, \dots, b_{22k}, I_{22}$	$b_{231}, b_{232}, \dots, b_{23k}, I_{23}$

Figure 1.3: Table of the parameters of the new Data





# Chapter 2

## A review of Symbolic Data Analysis

This chapter deal with principal components and clustering techniques. These methods are extensions of well-known classical theory applied or extended to symbolic data. Our approach has been to assume the reader is knowledgeable about the classical results, with the present focus on the adaptation to the symbolic data setting. In particular we will review these statistical methods for the two types of data presented in the previous chapter: interval data and histogram data.

### 2.1 Principal Components Analysis for Symbolic Data

Among the common data analysis method, Principal Components Analysis (PCA) is widely used to discover and to visualize the main structure of a multidimensional data set. A principal components analysis is designed to reduce  $p$ -dimensional observations into  $s$ -dimensional components (where usually  $s \ll p$ ). More specifically, a principal

component is a linear combination of the original variables, and the goal is to find those  $s$  principal components which together explain most of the underlying variance-covariance structure of the  $p$  variables. PCA takes as input a data matrix of the type  $X = (x_{ij})$ , where  $(x_{ij})$  is the precise and single value of the descriptive feature  $Y_j$  for the  $i$ -th object (for  $i = 1, \dots, n$ ). However, in practice the investigated objects are often more complex and so more complex data are required in order to provide an accurate description of these objects. These data are called *symbolic data*. The currently existing methods for performing principal components analysis on symbolic data as interval data and histogram data are covered in this section.

### 2.1.1 Principal Components Analysis of Symbolic Data described by intervals

Principal Components Analysis (PCA) aims to visualize, synthesize and compare units onto factorial spaces with minimum loss of information (for example minimum distortion of the distance between original data). Whereas units are represented by points, it is sufficient to just take care of their position in space. On the other hand, symbolic objects described by interval valued variables, represented as *boxes* in a multidimensional space, needs to be visualized, synthesized and compared onto the factorial spaces, taking care not only of their *location* but also by their *size* and *shape*. That is to say, two points can only be differentiated by their location in space, but two boxes can be differentiated also by their size (the volume of the box) and by their shape (a box can be narrow or wide in one or more dimensions compared to another box).

According to the Symbolic Data Analysis (SDA) paradigm, considering the input, the technique of analysis and the output, we may have two families of analysis: Symbolic(input)-Classical(treatment)-

Symbolic(output) and Symbolic-Symbolic-Symbolic.

The first family of analysis are historically the first introduced: they were based on a symbolic input table, a suitable numerical coding of data, a treatment with classical data analysis technique, a suitable transformation of classical results into a symbolic description. To this approach belongs Vertices PCA, Centers PCA and SPCA.

The first approach [11] to the treatment of multidimensional boxes considered a two step analysis based first on the numerical coding of a box vertices or its center and then performing a classic PCA on this coded data (Vertices PCA, Centers PCA).

A second approach [42], implemented in the SODAS/ASSO software, stresses the fact that a box is a cohesive set of vertices that also depends on its size and shape, introducing a more consistent way to treat units as complex data representation by introducing suitable constraints for the vertices belonging to the same object. This approach overcomes the drawback of the previous approach, where vertices are treated as single independent units described by points. Both the approaches propose to represent boxes on factorial plans as rectangles of minimum area enclosing the projections of vertices for each box. Such rectangles are interpreted as well as symbolic objects.

More recently, in order to avoid loss of information due to the data transformation, the intervals algebra introduced by R.E. Moore [47] is considered for a different approach to the boxes PCA.

Among the interval algebra theorems for the computation of interval data functions, one has been emphasized for the treatment of such kinds of data: “If a system of equations has a symmetric coefficient matrix, then the midpoints of the interval solutions are equal to the solution of the interval midpoints” [46]. This theorem permitted the development of new analysis methods not based merely on the representation of intervals by means of its extreme points (the vertices of the boxes), but based on codifying the intervals by its centre or *midpoints* and *radii*. In this direction, intermediate families of analy-

sis have been developed. Indeed, they work on a symbolic table as input, and classical techniques are extended to take into account some interval algebra theorems or definitions, the output is reconstructed symbolic data according to the same theorems of interval algebra. We called this family a *hybrid* approach since the treatment step is neither fully classical nor fully based on interval algebra.

We refer in particular to the methods called MRPCA (Midpoints Radii Principal Components Analysis, [48]), where classic linear algebra techniques are used to treat intervals coded as a pair (*midpoint*, *radius*). This is a hybrid approach in the sense that it takes into consideration some theorems of interval algebra but uses a classic linear algebra algorithm to analyse data just rebuilding boxes ex-post on the factorial plans.

In order to accomplish the Symbolic-Symbolic-Symbolic paradigm of analysis Gioia and Lauro (2006) proposed an approach, developed using interval linear algebra, called IPCA [34]. This approach is fully consistent with the interval nature of the descriptors of boxes, and performs a PCA of a interval correlation matrix allowing interval eigenvalues and eigenvectors with interval components.

### 2.1.2 Generalization of the Principal Components Analysis to Histogram Data

Nowadays we often need to perform data analysis (such as principal components, discriminant analysis, regression, multidimensional scaling, etc.) on enormous data sets, so large that it makes standard or classical analysis extremely difficult to implement and interpret. To overcome these difficulties it may be necessary and useful to aggregate the data into summary-type classifications or classes, where the number of classes is drastically smaller than the number of single in-

dividuals in the original data set.

For example suppose a study involves several cities (or regions, countries, etc.) classified by occupation, age and gender. It may be useful to merge the data for each region, retaining the identifying classifications of “occupation”, “age”, “income” and “gender”. We may wish to describe and analyze underlying concepts such as unemployment and we may also want to query the data set relating to the absence or presence of certain occupations. In these (and related examples) the aggregation process gives rise to symbolic data rather than classical data values on some if not all of the variables describing each symbolic object or observation of the aggregated data set. Most likely, symbolic data methods may have been an integral part of the aggregation procedure.

In 1997 Cazes, Chouakria, Diday and Schektman [11] proposed the Centers and the Tops Methods to extend the known principal components analysis method, PCA, to a particular kind of symbolic objects characterized by multi-valued variables of interval-type.

Subsequently Rodriguez, Diday and Winsberg [51] proposed an extension of classical PCA to interval data. Using the duality theorem they presented an improved algorithm for centers PCA and then they extended centers PCA to histogram-type including the case where the data are of mixed types, histogram, interval, classical (single-value) as well as the case where the data is of any one or two of these type of data.

To extend PCA to histogram type data they developed the idea first proposed in [25]. They represent each histogram-individual by a succession of  $k$  interval-individuals (the first one included in the second one, the second one included in the third one and so on) where  $k$  is the maximum number of modalities taken by some variable in the input symbolic data table.

Instead of representing the histograms in the factorial plane, they are going to represent the Empirical Distribution Function  $F_Y$  defined in

[7] associated with each histogram. In other words if we have a histogram variable  $Y$  on a set  $E = a_1, a_2, \dots$  of objects with domain  $\Upsilon$  represented by the mapping  $Y(a) = (U(a), \pi_a)$  for  $a \in E$ , where  $\pi_a$  is a frequency distribution, then in the algorithm they will use the function  $F(x) = \sum_{i/\pi_i \leq x} \pi_i$  instead of the histogram.

In particular, let  $X = (x_{ij})_{\substack{i=1,\dots,m \\ j=1,\dots,n}}$  be a symbolic data table with contin-

uous, interval and histogram variables types, and  $k = \max\{s, \text{where } s \text{ is the number of } m, 1, \dots, n \text{ where } Y^j \text{ is of histogram type}\}$ . They define the vector-succession of intervals associated with each cell of  $X$  as:

1. if  $x_{ij} = [a, b]$  then the vector-succession of intervals associated is:

$$x_{ij}^\downarrow = \begin{bmatrix} [a, b] \\ [a, b] \\ \vdots \\ [a, b] \end{bmatrix}_{k \times 1}$$

2. if  $x_{ij} = (1(p_1), 2(p_2), \dots, s(p_s))$  with  $s < k$  (histogram) then the vector-succession of intervals associated is:

$$x_{ij}^\downarrow = \begin{bmatrix} [0, p_1] \\ [0, p_1 + p_2] \\ \vdots \\ [0, \sum_{w=1}^s p_w] \end{bmatrix}_{k \times 1}$$

3. if  $x_{ij} = a$  then the vector-succession of intervals associated is:

$$x_{ij}^\downarrow = \begin{bmatrix} [a, a] \\ [a, a] \\ \vdots \\ [a, a] \end{bmatrix}_{k \times 1}$$

## 2.2. Clustering of Symbolic Data

---

As well, they defined the row-vector associated with each cell of  $X$  as:

1. if  $x_{ij} = [a, b]$  then the row-vector associated is:

$$x_{ij}^{\rightarrow} = \left[ \frac{a+b}{2} \right]_{1 \times 1}$$

2. if  $x_{ij} = (1(p_1), 1(p_2), \dots, s(p_s))$  where  $s$  is the number of modalities of the  $j$ -th variable, then the row-vector associated is:

$$x_{ij}^{\rightarrow} = [p_1, p_2, \dots, p_s]_{1 \times s}$$

3. if  $x_{ij} = a$  then the row-vector associated is:

$$x_{ij}^{\rightarrow} = [a]_{1 \times 1}$$

So, they apply the algorithm proposed in [50] to the matrix  $X^{\downarrow}$ . With this PCA they can find the shape of the “individual-histogram” in the principal plane. However since all the individual-histograms will be projected almost in the same position, they apply another PCA in order to find a good cluster structure to the individual-histogram. Therefore they apply a classical PCA to a matrix  $X^{\rightarrow}$ . Using this last principal component, they translate the individual-histogram to find the cluster structure of the individual-histogram in the principal plane.

## 2.2 Clustering of Symbolic Data

One of the common tasks in (classical as well as symbolic) data analysis is the detection and construction of “Homogeneous” groups  $C_1, C_2, \dots$  of objects in a population  $E$  as such an object from the same group

show a high similarity whereas objects from different groups are typically more dissimilar. Such groups are usually called “clusters” and must be constructed on the basis of the (classical or symbolic) data which were recorded for the objects. Cluster Analysis is a collective name for a range of mathematical, statistical or algorithmic methods for subdividing the total set  $E$  into homogeneous clusters which are typically compiled in a classification  $\mathcal{C} = (C_1, C_2, \dots)$ . The method can be classified according to various criteria such as: type of data, type of clustering criterion, type of classification structure, type of algorithm, etc. Since the clustering structures (partitioning, hierarchical, and pyramidal clustering) are based on dissimilarity measures, we will first describe these measures as they pertain to symbolic data, in the particular case when we have real valued-data represented by intervals and histograms.

### 2.2.1 Dissimilarity measures of Interval Data

The formation of subsets  $(C_1, \dots, C_r)$  of  $E$  into a partition, hierarchy, or pyramid is governed by similarity  $s(a, b)$  or dissimilarity  $d(a, b)$  measures between two objects, say  $a$  and  $b$ . These measures take a variety of forms. Since a similarity measure is typically an inverse functional of its corresponding dissimilarity measure (e.g.,  $s(a, b) = 1 - d(a, b)$ ), and the like), we consider just dissimilarity measures. Distance measures are important examples of dissimilarity measures.

**Definition.** A *dissimilarity measure*  $d$  on a set  $E$  is a function:

$$\begin{aligned} d : E \times E &\rightarrow \mathfrak{R}^+ \\ (k, l) &\rightarrow d(k, l) \end{aligned}$$

such as

- (i)  $d(k, l) = d(l, k) \ \forall (k, l) \in E \times E$
- (ii)  $d(k, k) = 0 \ \forall k \in E$



## 2.2. Clustering of Symbolic Data

---

**Definition.** A *distance measure* (also called a *metric*) is a dissimilarity measure which further satisfies:

$$(iii) \quad d(a, b) \leq d(a, c) + d(c, b) \quad \forall a, b, c \in E.$$

**Definition.** An *ultrametric measure* is a distance measure which also satisfies:

$$(iv) \quad d(a, b) \leq \text{Max}\{d(a, c), d(c, b)\} \quad \forall a, b, c \in E.$$

In this section, we assume that we have a given set of symbolic objects represented by the rows of a symbolic data array  $\underline{X} = (\xi_{kj})_{n \times p}$ . We want to extract detailed or global information from  $\underline{X}$  by special data analysis methods. In order to exploit the power of *classical* data analysis, such as multidimensional scaling, clustering, factorial or discriminant analysis, one approach consists in generating a classical dissimilarity or similarity matrix from the symbolic objects and applying the classical methods to this matrix. In literature, several dissimilarity measures have been proposed for interval data.

Gowda and Diday (1991) proposed a dissimilarity measure  $D(a, b)$  for two multidimensional interval (box)  $a = (A_1, \dots, A_p)$  e  $b = (B_1, \dots, B_p)$  dove  $A_j = [\underline{a}_j, \bar{a}_j]$ ,  $B_j = [\underline{b}_j, \bar{b}_j]$ .

This distance function is given in additive form:

$$D(a, b) = \sum_{j=1}^p D(A_j, B_j) \quad (2.1)$$

For the  $j$ -th variable,  $D(A_j, B_j)$  is obtained by considering three types of dissimilarity measures defined for pairs of subsets  $A_j, B_j$  and incorporating different aspects of “similarity”:

$$D(A_j, B_j) = D_p(A_j, B_j) + D_s(A_j, B_j) + D_c(A_j, B_j)$$

with the following specification:

- The component  $D_p$  (*position component*) indicates the relative positions of the two variable values on real line and it is defined as follows:

$$D_p(A_j, B_j) = \frac{|\underline{a}_j - \underline{b}_j|}{|\mu(D_j)|}$$

where  $|\mu(D_j)|$  denotes the length of the maximum interval of the  $j$ -th variable.

- The component  $D_s$  (*span component*) indicates the relative sizes of the variable values without referring to common parts between them. It is defined as follows:

$$D_s(A_j, B_j) = \frac{|l_a - l_b|}{l_s}$$

where  $l_a := |\bar{a}_j - \underline{a}_j|$ ;  $l_b := |\bar{b}_j - \underline{b}_j|$  and  $l_s = |\max(\bar{a}_j, \bar{b}_j) - \min(\underline{a}_j, \underline{b}_j)|$ .

- Finally, the component  $D_c$  (*content component*) is a measure of the noncommon parts between two variable values. It is defined as:

$$D_c(A_j, B_j) = \frac{l_a - l_b - 2\mu(A_j \cap B_j)}{l_s}$$

where  $\mu(A_j \cap B_j)$  is the length of intersection between  $A_j$  and  $B_j$ .

*Ichino and Yaguchi (1994)* proposed another dissimilarity measure between two symbolic objects  $a$  and  $b$  of the type (2.1). First, they defined two Cartesian operators, *join*  $\oplus$  and *meet*  $\otimes$ , which were applied to the pairs of subsets  $(A_j, B_j)$ :

$$A_j \oplus B_j := [\min(\underline{a}_j, \underline{b}_j), \max(\bar{a}_j, \bar{b}_j)]$$

$$A_j \otimes B_j := A_j \cap B_j.$$

## 2.2. Clustering of Symbolic Data

---

Now it is possible to define the Ichino & Yaguchi's dissimilarity measure:

$$\phi(A_j, B_j) := |A_j \oplus B_j| - |A_j \otimes B_j| + \gamma(2 \cdot |A_j \otimes B_j| - |A_j| - |B_j|)$$

where  $0 \leq \gamma \leq 0.5$  is a prespecified parameter and  $|A_j|$  denotes the length of the interval  $A_j$ . The parameter  $\gamma$  plays an important role in this definition, in fact it controls the effect of the inner-side nearness and outer-side nearness between  $A_j$  and  $B_j$  on the distance.

It is possible to define the *generalized Minkowski distance* of order  $q$  ( $q \geq 1$ ) as:

$$d_q(a, b) = \left( \sum_{j=1}^p \phi(A_j, B_j)^q \right)^{1/q} \quad (2.2)$$

where all the variable  $Y_j$  may be expressed with different units of measurements. A normalized formulation for  $\phi$  is:

$$\psi(A_j, B_j) = \frac{\phi(A_j, B_j)}{|\mu(D_j)|}. \quad (2.3)$$

De Carvalho (1994,1996,1998) proposes an extensions of the previous dissimilarity measures of Ichino and Yaguchi that concerns the function  $\psi$  and  $\phi$  which were combined by Ichino and Yaguchi into the generalized Minkowski metric. De Carvalho combines several function, called *comparison functions (CF)*, with an *aggregation function (AF)*, such as Minkowski's metric. He suggests the calculation of comparison function of each variable  $Y_j$  on the basis of the agreement indices summarized in the figure 2.1.

De Carvalho has proposed the comparison functions, see figure 2.2 as an extension of the similarity measures defined for classical binary variables. (Note that each similarity function generates a corresponding dissimilarity function).

	Agreement	Disagreement	Total
Agreement	$\alpha = \mu[A_j \cap B_j]$	$\beta = \mu[A_j \cap c[B_j]]$	$\mu[A_j]$
Disagreement	$\gamma = \mu[c[A_j] \cap B_j]$	$\delta = \mu[c[A_j] \cap c[B_j]]$	$\mu[c[A_j]]$
Total	$\mu[B_j]$	$\mu[c[B_j]]$	$\mu[D_j]$

Figure 2.1: Table agreement indices

As for the distance  $\psi$  and  $\phi$  of Ichino and Yaguchi's proposal, De Carvalho selects, for each component variable  $Y_j$ , one of the dissimilarities  $d_i$  and combine them with an *aggregation function*  $f$  as with Minkowski's metric. This results in the overall dissimilarity:

$$d_a^i(a, b) = \sqrt[q]{\sum_{j=1}^p [w_j d_i(A_j, B_j)]^q} \quad (2.4)$$

with  $i \in \{1, \dots, 5\}$ .

De Carvalho (1996) also proposed another comparison function  $\psi'$  in combination with an appropriate aggregation function  $f$ . This comparison function is defined as a further normalization of Ichino and Yaguchi's  $\phi$  function:

$$\psi'(A_j, B_j) := \frac{\phi(A_j, B_j)}{\mu(A_j \oplus b_j)}. \quad (2.5)$$

*Souza and De Carvalho (2004)* proposed the "city-block" distance: Let two vectors of intervals:  $a = (A_1, \dots, A_p)$  and  $b = (B_1, \dots, B_p)$  where  $A_j = [\underline{a}_j, \bar{a}_j]$ ,  $B_j = [\underline{b}_j, \bar{b}_j]$ .

The city-block distance is defined as:

$$d(a, b) = \sum_{j=1}^p \phi(A_j, B_j) = \sum_{j=1}^p [|\underline{a}_j - \underline{b}_j| + |\bar{a}_j - \bar{b}_j|].$$

## 2.2. Clustering of Symbolic Data

---

$s_c^j$	Comparison function	Range	Property $\phi_c^j = 1 - s_c^j$	= 0 if	= 1 if
$s_c^1$	$\frac{\alpha}{\alpha + \beta + \gamma}$	[0, 1]	Metric	$A_j \cap B_j = \emptyset$	$A_j = B_j$
$s_c^2$	$\frac{2\alpha}{2\alpha + \beta + \gamma}$	[0, 1]	Semi Metric	$A_j \cap B_j = \emptyset$	$A_j = B_j$
$s_c^3$	$\frac{\alpha}{\alpha + 2(\beta + \gamma)}$	[0, 1]	Metric	$A_j \cap B_j = \emptyset$	$A_j = B_j$
$s_c^4$	$\frac{1}{2} \left[ \frac{\alpha}{\alpha + \beta} + \frac{\alpha}{\alpha + \gamma} \right]$	[0, 1]	Semi Metric	$A_j \cap B_j = \emptyset$	$A_j = B_j$
$s_c^5$	$\frac{\alpha}{\sqrt{(\alpha + \beta)(\alpha + \gamma)}}$	[0, 1]	Semi Metric	$A_j \cap B_j = \emptyset$	$A_j = B_j$

Figure 2.2: Table of comparison function

This distance function is a suitable extension of the  $L_1$  metric to interval data.

*Chavent and Lechevallier (2002)* proposed the *Hausdorff distance* defined as:

$$d_H(A, B) = \max \left( \sup_{x \in A} \inf_{y \in B} d(x, y), \sup_{y \in B} \inf_{x \in A} d(x, y) \right). \quad (2.6)$$

If  $d(x, y)$  is the  $L_1$  *City block* distance, then *Chavent et al. (2002)* proved that:

$$d_H(a, b) = \max \left\{ |\underline{a}_j - \underline{b}_j|, |\bar{a}_j - \bar{b}_j| \right\}$$

In 2006 *De Carvalho et al* proposed a family of distance between intervals, the metric of norm  $q$  defined as:

$$d_{L_q}(A, B) = \left( \sum_{j=1}^p |\underline{a}_j - \underline{b}_j|^q + |\bar{a}_j - \bar{b}_j|^q \right)^{1/q}. \quad (2.7)$$

In particular for  $q = 2$  we have the "*Squared Euclidean*" distance:

$$d(a, b) = \sum_{j=1}^p \phi(A_j, B_j) = \sum_{j=1}^p \left[ (\underline{a}_j - \underline{b}_j)^2 + (\bar{a}_j - \bar{b}_j)^2 \right].$$

*Antonio Irpino and Rosanna Verde (2006)* proposed the *Wasserstein* distance: If we suppose a uniform distribution of points, an interval of reals  $A(t) = [a, b]$  can be expressed as the following type of function:

$$A(t) = [a, b] = a + t(b - a) \quad 0 \leq t \leq 1.$$

If we consider a description of interval by means of its midpoint  $m$  and radius  $r$ , the same function can be rewritten as follows:

$$A(t) = m + r(2t - 1) \quad 0 \leq t \leq 1.$$

Then, the squared Euclidean distance between homologous points of two intervals  $A = [\underline{a}, \bar{a}]$  and  $B = [\underline{b}, \bar{b}]$ , or described by the midpoint-radius notation  $A = (m_A, r_A)$  and  $B = (m_B, r_B)$ , is defined as follows:

$$\begin{aligned} d_W^2 = (A, B) &= \int_0^1 [A(t) - B(t)]^2 dt = \\ &= \int_0^1 [(m_A - m_B) + (r_A - r_B)(2t - 1)]^2 dt = \\ &= (m_A - m_B)^2 + \frac{1}{3}(r_A - r_B)^2. \end{aligned}$$

### 2.2.2 Distance measures between histogram data

In order to cluster a set of data described by distribution with finite continue support, or, as called in SDA, by "histograms" we have to define a distance between them.

A set of metrics, defined in probability measure spaces, seems particularly interesting to measure the probability between distributions. So,

## 2.2. Clustering of Symbolic Data

---

they can be proposed in the clustering analysis when data are considered as (empirical) distributions. These metrics were born in the framework of convergence theory. See figure 2.3

Gibbs and Su [32] present a good review on metrics between probability measures (histograms can be considered as the representation of empirical frequency distribution).

Given a domain  $\Omega$  on which it is possible to define a Borel  $\sigma$ -algebra  $\mathcal{B}$ , two measures  $\mu$  and  $\nu$  (like  $\pi_{ih}$  are) on  $\Omega$ ,  $f$  and  $g$  the density function with respect to a  $\sigma$ -finite dominant measure  $\lambda$ .  $F$  and  $G$  denote the corresponding distribution functions. Gibbs and Su [32] present a review of the most used dissimilarities; see figure 2.4.

Abbreviation	Metric
D	Discrepancy
H	Hellinger distance
I	Relative entropy (or Kullback-Leibler divergence)
K	Kolmogorov (or Uniform) metric
L	Lévy metric
P	Prokhorov metric
S	Separation distance
TV	Total variation distance
W	Wasserstein (or Kantorovich) metric
$\chi^2$	$\chi^2$ distance

Figure 2.3: Metrics and their abbreviation

In a different context of analysis, Chavent et al. ?? propose two measure for the comparison of histograms: the  $L^2$  norm and a *two component* dissimilarity.  $L^2$  norm is simply computed considering the weights of the elementary intervals but not their width. While the *two component* is a dissimilarity which does not satisfy the usual properties of distance measures.

## A REVIEW OF SYMBOLIC DATA ANALYSIS

---

Metric	State space	Definition	Image	Remarks
<b>Discrepancy</b>	Any metric space	$d_D(\mu, \nu) := \sup_{\text{all closed balls } B}  \mu(B) - \nu(B) .$	$[0; 1]$	
<b>Hellinger</b>	Any measurable space	$d_H(\mu, \nu) := \left[ \int_0^1 (\sqrt{f} - \sqrt{g})^2 d\lambda \right]^{1/2} = \left[ 2 \left( 1 - \int_0^1 \sqrt{f g} d\lambda \right) \right]^{1/2}.$	$[0; 2^{1/2}]$	
<b>Rel. Entropy (Kullback-Leibner)</b>	Any measurable space	$d_I(\mu, \nu) := \int_{S(\mu)} f \log(f/g) d\lambda.$	$[0; \text{Inf}]$	Is not a distance
<b>Kolmogorow (Uniform)</b>	$\mathbb{R}$	$d_K(F, G) := \sup_x  F(x) - G(x) , \quad x \in \mathbb{R}.$	$[0; 1]$	
<b>Levy</b>	$\mathbb{R}$	$d_L(F, G) := \inf\{\epsilon > 0 : G(x - \epsilon) - \epsilon \leq F(x) \leq G(x + \epsilon) + \epsilon, \forall x \in \mathbb{R}\}.$	$[0; 1]$	Not easy to compute

Metric	State space	Definition	Image	Remarks
<b>Prokhorof</b>	Any metric space	$d_P(\mu, \nu) := \inf\{\epsilon > 0 : \mu(B) \leq \nu(B^\epsilon) + \epsilon \text{ for all Borel sets } B\}$ where $B^\epsilon = \{x : \inf_{y \in B} d(x, y) \leq \epsilon\}$	$[0; 1]$	
<b>Separation distance</b>	Any countable space	$d_S(\mu, \nu) := \max_i \left( 1 - \frac{\mu(i)}{\nu(i)} \right).$	$[0; 1]$	Is not a distance (not symmetric)
<b>Total variation distance</b>	Any measurable space	$d_{TV}(\mu, \nu) := \sup_{A \subset \Omega}  \mu(A) - \nu(A) $ where $h : \Omega \rightarrow \mathbb{R}$ satisfies $ h(x)  \leq 1$ .	$[0; 1]$	
<b>Wasserstein or Kantorovich metric</b>	Any measurable space	$d_W(\mu, \nu) := \int_0^1  F^{-1}(t) - G^{-1}(t)  dt.$	$[0; \text{Inf}]$	
<b><math>\chi^2</math>-distance</b>	Any measurable space	$d_{\chi^2}(\mu, \nu) := \int_{S(\mu) \cup S(\nu)} \frac{(f - g)^2}{g} d\lambda.$	$[0; \text{Inf}]$	Is not a distance (not symmetric)

Figure 2.4: Metrics and their definitions



# Chapter 3

## Model Data Building

The idea of using a model to represent a histogram data comes from the need of working with functions, instead of empirical values, that could smooth a histogram and thus the possibility to leave out outlier values. Everything is based on the known paradigm:

$$DATA = MODEL + ERROR.$$

Therefore we transform the data represented by a histogram to a model that synthesizes the shape of distribution with a certain error, obviously depending on the kind of approximation. We are looking for the best trade-off between model and error.

The aim is to deal with comparable models in order to use Multidimensional Data Analysis. Therefore all the histograms will be transformed in models by means of approximations with functions belonging to the same family.

The best trade-off concerns the choice of the model, or better the choice of the number of parameters to use in the approximation, and the error due to the approximation. In a trivial case, it is possible to achieve a model that perfectly fits the histogram shape in order to get

an approximation error equal to zero but with a huge number of parameters (i.e. the number of histogram bins). In that case the procedure would be useless because we would simply be dealing with the frequencies of the related histogram classes. The idea is to get a smaller fixed number of parameters a priori equal for all the histograms and thus to obtain different approximation errors. So the best arrangement is to find the number of parameters to fix.

We are supposed to find the number of parameters as well as about half the number  $k$  of the histogram classes fixing that value inside the interval  $\left[\text{int}\left(\frac{k}{2}\right) - 1, \text{int}\left(\frac{k}{2}\right) + 1\right]$ , and later work out a suitable index of goodness of fit that allows us to know the approximation quality. Therefore, the first step is to choose a suitable function for the approximation procedure. We could approach with the basis functions and that way manage to find out the one that better fits the problem. In this chapter the definition of mathematical model is introduced differentiating the interpolation models from the approximation ones and some of the basis functions are described. Later we will see how to build the model.

### 3.1 Mathematical Model

Formulating a mathematical model means to determine a function  $f$ , in an interval  $I$ , in a way that is:

- represents data  $(x_i, y_i)$ ;
- preserves eventual correlation properties of the magnitudes;
- allows us to get new eventual requested information.

In order to work out a model that satisfies such requirements, indispensable for a reliable model, it is necessary to take care of the rules that control the phenomenon. That information can help to define

### 3.1. Mathematical Model

---

the shape of the model, that is the kind of function  $f$  (e.g. a rect, a parabola, a trigonometrical function, etc.).

Then we can give the following definition:

**Definition.** Given a finite set of data  $D = \{(x_i, y_i)_{i=1, \dots, n}\}$  belonging to the interval  $I$ , each function  $f$ , defined on  $I$  that describe  $D$ , is called a *fitting* or *model* for  $D$ ; then such function is called *interpolation function* (or *approximation function*) according to verify (or not) the conditions of the function and/or its derivatives in the assigned points, that is to satisfy (or not) the condition:

$$f(x_i) = y_i \quad (\text{generally } f^{(j)}(x_i) = y_i^j \quad j = 0, \dots, m-1) \quad \forall i = 1, \dots, n.$$

called *interpolation conditions*.

So to determine a model it is, first of all, necessary to establish if  $f$  has to approximate or interpolate the data. In the approximating case it is also necessary to be able to set a measure of how far  $f$  is from the points  $D$ . The use of an interpolation model strictly bound by the data makes sense only when the data is not affected by *negligible* errors. On the contrary, if the data is affected by *not negligible* errors, it would not make sense to bind a function and assume those values because it could amplify the error.

**Definition (Interpolation problem).** Given  $n$  different values  $(x_i)_{i=1, \dots, n}$  called knots, and  $n$  corresponding values  $(y_i)_{i=1, \dots, n}$ , we want to determine a function  $f$  called *interpolation function*, that on the knots  $(x_i)_{i=1, \dots, n}$  satisfies certain conditions, called *interpolation conditions*. Such conditions, generally, are constraints that the interpolation function  $f$  (and/or its derivatives), must satisfy on the points  $(x_i, y_i)_{i=1, \dots, n}$ .

**Definition (Approximation problem).** Given  $n$  different values  $(x_i)_{i=1, \dots, n}$  called knots, and  $n$  corresponding values  $(y_i)_{i=1, \dots, n}$  we want to determine a function  $f$  called *approximation function*, in a way that the distance between  $f(x_i)$  and  $y_i$  is minimum; the choice of the mea-

sure of such a distance qualifies the approximation problem.

Interpolation and approximation provide two basically different models, even if historically they have been confusing. One of the reasons is that among the approximation functions we could choose the ones that in some knots  $x_i$  assume the values  $y_i$ , in other words we could choose to work out functions that can also be interpolation functions.

## 3.2 Basis functions approach

A basis function is an element of the base of a function space. In such a space, each function can be represented as a linear combination of basis functions.

The well-known basis functions are:

- Polynomial basis
- Piecewise Polynomial basis
- Spline
- B-spline

When the number of interpolation points is high the polynomial does not normally provide a reliable model because the higher the number of points, the higher the interpolation polynomial degree, the polynomial oscillations would also increase getting into a not always consistent model with the points trend.

In order to decrease the polynomial degree and its oscillations, partitioning the knots interval in contiguous sub-intervals and building the interpolation model locally, that is on each sub-interval, could be a good strategy. That allows us to determine the lowest degree interpolation polynomial independently from how many knots we have.

Anyway, the eventual discontinuities on the connections between polynomials are not avoided so in order to determine a more satisfying model it is desirable to be able to construct:

- a low polynomial degree;
- an smooth enough function along the whole interval.

The solution to this problem is provided by particular piecewise polynomial functions called *spline*.

#### 3.2.1 Polynomial basis functions

Polynomial basis functions are used for approximating because they can be evaluated, differentiated, and integrated easily and in finitely many steps using just the basic arithmetic operations of addition, subtraction and multiplication.

A polynomial of order  $n$  or of degree  $n - 1$  is a function of the form

$$p(x) = a_1 + a_2x + \dots + a_nx^{n-1} = \sum_{j=1}^n a_jx^{j-1}$$

Although polynomials represent a flexible way to represent a function for their easy computation, they have limited to appeal due to their local nature, that depends on the choice of interpolation points, and they have some problems in approximating the function on the extremes of large intervals. Moreover, the increase of the polynomial degree does not necessary imply a better representation of the function, often, the contrary, strong oscillations arise. The situation may be improved in different ways. One of this is to keep the polynomial degree fixed, to split the intervals of interest into smaller pieces and consider functions which are continuous on the defined intervals. Such kind of approximation is called piecewise approximation. It is

more flexible and it allows us to avoid large oscillation observed for high-degree polynomial approximation.

### 3.2.2 Piecewise Polynomial Basis

Assume that the interval  $[a, b]$  is split into  $M$  segments by a sequence of points  $t = \{t_m\}_{m=1}^M$  such that  $[a \leq t_1 \leq t_2 \leq \dots \leq t_M \leq b]$  the piecewise polynomial function can be defined as following:

Let  $H$  be a positive integer. The corresponding piecewise polynomial function  $f(t) \in P_{H,t}$  (the space of polynomial order  $H$  with knots  $t$ ) of order  $H$  (degree  $H - 1$ ) is defined by:

$$f(t) = \sum_{l=0}^{H-1} \alpha_l t^l \chi \{t \in [t_m, t_{m+1}]\}$$

where  $\chi \{t \in [t_m, t_{m+1}]\}$  is the indicator function defined on each subinterval  $[t_m, t_{m+1}]$ .

The defined polynomial is obtained by a separate polynomial in each interval, in term of basis functions it can be written as:

$$f(t) = \sum_{m=1}^M \sum_{l=1}^H \alpha_{lm} \phi_{lm}$$

where  $\phi_{lm} = \Xi \{t \in [t_m, t_{m+1}]\}$ . The main disadvantage of this definition is that the piecewise polynomial functions are not continuous or smooth in the interior knots, this problem can be achieved by imposing the following continuity conditions

$$f_{m-1}(t_l) = f_m(t_l) \quad m = 1, \dots, M$$

and

$$D^l f_{m-1}(t_h) = D^l f_m(t_h) \quad m = 1, \dots, M \quad l = 1, \dots, H - 2.$$

A more direct way to proceed is to use a basis that incorporates these constraints, the so called *polynomials splines*.

### 3.2.3 Spline functions

**Definition.** Given the knots:

$$x_1 < x_2 < \dots x_n,$$

a *cubic spline function* defined on the knots set  $K = \{x_1, \dots, x_n\}$  is a function  $s(x)$  such that:

- $s(x) \equiv p(x) \in \prod_3 \forall x \in [x_i, x_{i+1}]$ ,  $i = 1, \dots, n - 1$ ;
- $s(x), s'(x), s''(x)$  are continuous functions on the interval  $[x_1, x_n]$ .

Given the points with coordinates  $(x_i, y_i)_{i=1, \dots, n}$ , for building a interpolation cubic spline function we work out a polynomial that represents the spline in each interval. On each interval between the consecutive knots  $[x_1, x_2], \dots, [x_{n-1}, x_n]$  the spline function  $s$  must be a polynomial with at most 3 degrees:

$$s(x) \equiv p^i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \in \prod_3 x \in [x_i, x_{i+1}], i = 1, 2, \dots, n-1.$$

On each interval, then, it is required to determine four coefficients  $a_i, b_i, c_i, d_i$  and so altogether  $4(n - 1)$  coefficients.

In general, a *spline function* with degree  $p$  (or with order  $p + 1$ ), defined on a subset  $[a, b] \subseteq \mathfrak{R}$ , is a piecewise polynomial function.

The set  $[a, b]$  is partitioned in intervals  $[P_1, P_2, \dots, P_{k-1}]$  and  $t = [t_1 < t_2 < \dots < t_k]$  is a ordered knots subset or separation points of those intervals.

The number  $K$  of necessary knots so that the function is defined depends on the degree  $p$  of the function, that is  $K > p + 1$ .

A spline function  $s(p, t)$  takes advantages of the following properties:

- on each interval  $[t_k, t_{k+1}]$  ( $k = 1, \dots, K - 1$ ), is a polynomial of degree at most  $p$ ;

- allows derivatives up to order  $p - 1$ ;
- its derivatives are continuous on the interval  $[a, b]$ .

Every spline function of degree  $p$  on a set of knots  $t$  is unequivocally determined by:

$$s(p, t) = \sum \alpha_k (z - t_k)_+^p \quad z \in [t_k, t_{k+1}]$$

where  $\alpha_k$  are the real constants. The function  $(z - t_k)_+^p$  is called truncated power basis  $p$ .

The truncated power basis are piecewise polynomials and represent bases for building spline function. The computation of spline function by means of truncated power functions requires the definition of  $K + p - 1$  conditions.

In order to decrease the number of operations it could be worth using a different generation base, such that the related functions are still spline functions. This leads to the definition of *B-spline*.

### 3.2.4 B-spline

The B-spline functions of degree  $p$  compose a base in the subspace of all the spline functions of degree  $p$ . Actually, a spline function of degree  $p$ , defined on a knots set  $\{t_k\}_{k=0, \dots, n}$ , can be expressed as a linear combination of B-spline functions  $B_{i,p}$  on the knots sequence  $\{t_k\}_{k=0, \dots, n}$ :

$$S(t) = \sum_{i=0}^m P_i B_{i,p}(t)$$

where  $P_i$  are  $m + 1$  control points,  $\{t_0, \dots, t_n\}$  a knots sequence,  $p$  is the polynomial degree, and the functions  $B_{i,p}$  are the so called B-spline



### 3.2. Basis functions approach

---

functions that are built in the following way:

$$\begin{aligned} B_{i,1}(t) &= \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B_{i,p}(t) &= \frac{t-t_i}{t_{i+p-1}-t_i} B_{i,p-1}(t) + \frac{t_{i+p}-t}{t_p-t_{i+1}} B_{i+1,p-1}(t) \end{aligned}$$

Therefore, a B-spline curve involve more data: a set of  $m + 1$  control points, a vector of  $n + 1$  knots and a degree  $p$ , and for them the following expression must agree with:  $n = m + p + 1$ . To be more accurate, if we want to define a B-spline curve of degree  $p$  with  $m + 1$  control points, we will provide  $m + p + 2$  knots. On the other hand, if a vector of  $n + 1$  knots and  $m + 1$  control points are given, the degree of the B-spline curve will be  $p = n - m - 1$ . The B-spline functions exploit the following properties:

1. Partition of unity:

$$\sum_i B_{i,p} = 1$$

2. No negativity:

the B-spline function are always no negative, in fact:

$$B_{i,p}(z) = \begin{cases} > 0 & t_i \leq z \leq t_{i+p+1} \\ 0 & \text{otherwise} \end{cases}$$

3. Local support:

each integer  $t_i$  can give univocally a B-spline function of degree  $p$  whose support is the interval  $[t_i, t_{i+p+1}]$ , that is, the function is null outside that interval.

4. Recurrence:

B-spline functions of higher order can be obtained by the ones of lower order through the recurrence formula:

$$B_{i,p}(t) = \frac{t-t_i}{t_{i+p-1}-t_i} B_{i,p-1}(t) + \frac{t_{i+p}-t}{t_p-t_{i+1}} B_{i+1,p-1}(t)$$

The B-spline curve  $S(t)$  is a way to represent a curve through each component of a  $p$  degree curve. It takes advantage of the Strong Convex Hull property: a B-spline curve is included in the convex domain of its control polygon. Precisely, if  $t \in [t_i, t_{i+1})$ , then  $S(t)$  is in the convex hull of the control points  $P_{i-p}, P_{i-p+1}, \dots, P_i$ . If  $t \in [t_i, t_{i+1})$ , there are just  $p+1$  not null base functions (that is,  $B_{i,p}(t), \dots, B_{i-p+1,p}(t), B_{i-p,p}(t)$ ) on that interval. Since  $B_{k,p}(t)$  is the coefficient of the control point  $P_k$ , just  $p+1$  control points  $P_i, P_{i-1}, P_{i-2}, \dots, P_{i-p}$  do not have null coefficients. Since on that interval the base functions are not null and their sum is 1, their "weighted" average,  $S(t)$ , must lie on the convex domain defined by the control points  $P_i, P_{i-1}, P_{i-2}, \dots, P_{i-p}$ . The meaning of "strong" comes from the fact that  $S(t)$  lies on the smallest convex domain.

Generally the B-spline functions do not go through the control points, so neither through the starting and final point of the control polygon. Nevertheless, if a knots has multiplicity is equal to the degree  $p$  of the curve, the curve will pass through that point. For example, in order to make a cubic B-spline function pass through the starting point, it is necessary to collapse the first three knots: the knots sequence will start with  $0, 0, 0$ .

The B-spline curve exploits the property of Local Change: changing the position of the control points  $P_i$  affects the curve  $S(t)$  only in the interval  $[t_i, t_{i+p+1})$ . This comes from another important property of the base B-spline functions. We must remember that  $B_{i,p}(t)$  is a not null function in the interval  $[t_i, t_{i+p+1})$ . If  $t$  is not in that interval,  $B_{i,p}(t)$  does not affect at all the computation of  $S(t)$  because  $B_{i,p}(t)$  is null. This local change scheme is very important in drawing curves, because we are able to modify a curve locally without changing the global shape.

Furthermore, another important property is the Affine Invariance: if an affine transformation is applied to a B-Spline curve, the result can be achieved by the affine image of its control points. This is a useful

property when we want to apply a geometrical transformation or even affine transformation to a B-spline curve, because it establishes that we can apply the transformation to the control point, which is very easy, and once we have the control points transformed we have the transformed B-spline that is the only one defined by those new points. Then, we do not have to transform the curve.

In short, a B-spline function is basically a spline function with all its properties and furthermore all the previous comments done up to now are valid. It is possible to build a spline function and transform it in a B-spline, and vice versa. The difference lies in the expression of the curve. In the spline case we would deal with pp-form, or better, polynomial piecewise form while in the B-spline case we would deal with b-form. In the first case we will have so many coefficients as the spline function order for each polynomial piece. In the other case the number of control points will be established by the relation  $n = m + p + 1$ . For our purposes we liked using the B-spline function more because with an equal number of knots we have a smaller number of parameters of the model.

### 3.3 How to approximate a histogram using a B-spline

We would like to derive a smoother approximation from this histogram to the underlying distribution. We can do this by constructing a spline function  $f$  whose average value over each bar interval equals the height of that bar.

If  $h$  is the height of one of these bars, and its left and right edge are at  $L$  and  $R$ , then we want our spline  $f$  to satisfy

$$\left( \int_L^R f(x) \right) / (R - L) = h$$

or, with  $F$  the indefinite integral of  $f$ , i.e.,  $DF = f$ ,

$$F(R) - F(L) = h \cdot (R - L)$$

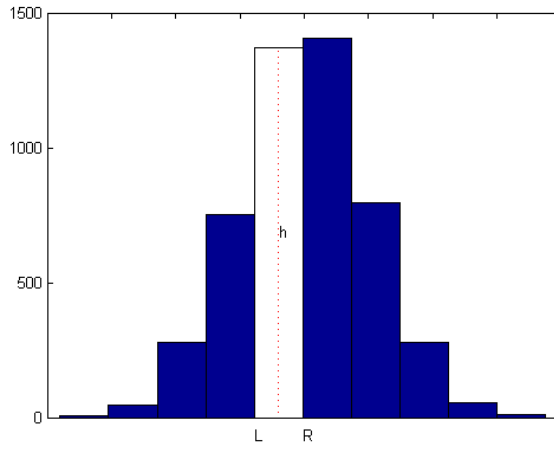


Figure 3.1: Histogram representation

So, with  $t(i)$  the left edge of the  $i$ -th bar,  $dt(i)$  its width, and  $h(i)$  its height, we want

$$F(t(i+1)) - F(t(i)) = h(i) \cdot dt(i), \quad i = 1, \dots, n,$$

or, setting arbitrarily  $F(t(1)) = 0$ ,

$$F(t(i)) = \sum_{j=1}^{i-1} (h(j) \cdot dt(j)), \quad i = 1, \dots, n+1.$$

### 3.3. How to approximate a histogram using a B-spline

---

Add to this the two end conditions  $DF(t(1)) = 0 = DF(t(n+1))$ , and we have all the data we need to get  $F$  as a complete cubic interpolant spline, and its derivative,  $f = DF$ , is what we want and plot, all in one statement, (see figure 3.2).

Since our purpose is to be able to compare different histograms, we

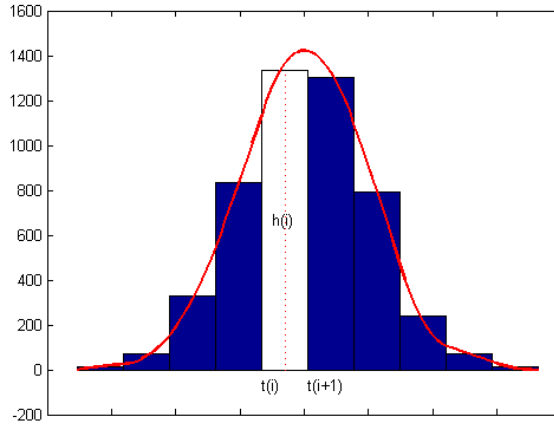


Figure 3.2: Smoothing an Histogram by spline

are going to transform the obtained spline functions, according to the basis function approach, in B-spline function.

What we get are as many control points as the number of histogram bars. Since the relation  $m = n - p - 1$  holds, having fixed  $p = 2$  and having imposed that the curve have to pass through starting and final points, (i.e. the knots sequence is  $\{t_0, t_0, t_1, \dots, t_n - 1, t_n, t_n\}$ ), we will have  $m = n - 1$ . In that case the obtained spline function is perfectly adapting the histogram.

A way to reduce the number of control points is to decrease the number of knots and/or to increase the spline function degree.

The problem is how to find an optimal knots sequence.

The number of knots and the degree of B-spline is chosen low in order to avoid overfitting and to have a parsimonious representation of histogram data. The degree of the B-spline usually do not exceed 3 and the location of knots should be established in terms of the best fitting function according to some experimental alternative number of knots, for example among 3 and 7 depending from the number of classes.

The idea is, to build a spline function starting from a given number of knots so that it can adjust the histogram in the best way. In the previous case we saw that the spline function fits perfectly according to the condition where in each bar the area is equal to the area down to the spline function in the same interval. This produces that the difference between those two quantities is equal to zero, therefore if we regard that difference as error size we could say that we have a null error and thus a perfect adaptation. But our purpose is not the perfect adaptation to the histogram because in that way we would aim to think of an empirical distribution that include the error term. We want to obtain a spline function that approximates the histogram except an error term (approximation error). So the starting point is to define a way to compute the error. By the previous comments we can consider measuring the error as the sum of the squares of the differences between the histogram bar area and the spline function area. In this way we build the objective function to minimize in order to find out the optimal sequence of knots. So we have to solve the following bound-constrained optimization problem:

$$\begin{aligned}
 & \underset{s.t.}{\operatorname{argmin}} \quad \sum_{h=1}^H \int_{L_i}^{R_i} [s(t) - h(i)]^2 dt \\
 & \quad t_0 \leq t \leq t_n \\
 & \quad |t_i - t_{i+1}| > (R_i - L_i)
 \end{aligned} \tag{3.1}$$

where  $L_i$ ,  $R_i$  and  $h(i)$  are respectively left edge, right edge and height of the  $i$ -th bar.

### 3.4. Histogram transformation process

---

In that way we are going to get a different knots sequence for each histogram and for this reason the B-spline parameters will not be comparable.

Since we can not determine an optimal sequence for every histogram, the idea is to create a knots sequence as the average of the obtained knots in each histogram. Later we will build an approximation spline function for each histogram starting from the knots mean sequence. So we will have comparable parameters of B-spline for each variable and so we could use them for following calculation.

## 3.4 Histogram transformation process

The starting point of trasformation process is a single value units $\times$ variables matrix where each unit is observed in  $N$  occasions. Starting from this, it is possible to build histograms by pooling occasions. So a new matrix is obtained where in each column there is an histogram variable, (see figure 3.3).

The step of the process are summarized as follow:






	VarY1	VarY2	...	VarYp
OS1			...	
...	...			
OSn		...		

Figure 3.3: Matrix of Histogram Data

- Step 1: Histograms building.

We want to obtain standardized histograms with the same number of bars of the same width. Regarding the number of classes the Strugres formula (1926) has been used:  $K = 1 + \log_2 N$  where  $N$  is the number of occasions. Moreover, to have a comparison among histogram we transform the histogram in  $[0, 1]$  by means:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.2)$$

where  $x$  is the occasions vector.

In this way we have built histograms with the same bins  $\{t_1, \dots, t_{K+1}\}$  (figure 3.4).

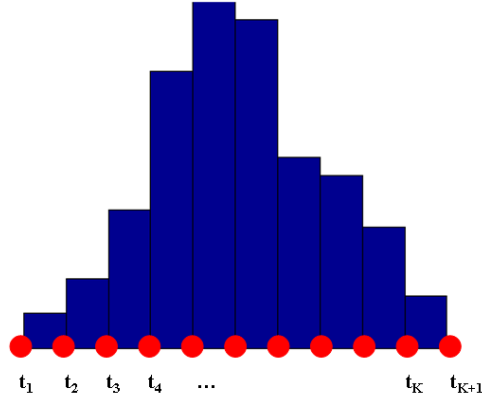


Figure 3.4: Bins sequence

- Step 2: Choose the number of optimal knots.  
Let us choose  $\{t_k\}_{k=1, \dots, K+1}$  as knots sequence, with  $t_1 = 0$  and



### 3.4. Histogram transformation process

---

$t_{K+1} = 1$ , and let use them as starting point to achieve the optimal sequence of knots. As said before we want to obtain a smaller number of knots within  $\left[ \text{int} \left( \frac{k}{2} \right) - 1, \text{int} \left( \frac{k}{2} \right) + 1 \right]$ . Two knots are fixed to the extremities, while the others are chosen by the optimal process (3.1) inside the interval  $(0, 1)$ . To simplify, from now on, we suppose to work on five knots.

So, we obtain three significant values for each histogram (the other two are equal for all the histograms) that is a matrix as in figure 3.5:

	VarY <sub>1</sub>	VarY <sub>2</sub>	...	VarY <sub>p</sub>
OS <sub>1</sub>	$t_{11}^1, t_{11}^2, t_{11}^3$	$t_{12}^1, t_{12}^2, t_{12}^3$		$t_{1p}^1, t_{1p}^2, t_{1p}^3$
...				
OS <sub>n</sub>	$t_{n1}^1, t_{n1}^2, t_{n1}^3$	$t_{n2}^1, t_{n2}^2, t_{n2}^3$		$t_{np}^1, t_{np}^2, t_{np}^3$

Figure 3.5: Matrix of the knots sequence

- Step 3: B-spline building.  
The B-spline are constructed starting from the optimal knots sequence  $t$  by means of

$$s(t) = \sum P_i B_i(t).$$

Our purpose is to compare the control points  $P_i$  and so the  $B_i(t)$  must be built on the same knots sequence to get the same bases.

- Step 4: Average knots sequence computation.  
In order to get the same knot sequence for each histogram variable, the idea is to build a mean vector of the knots calculating the mean for each column. (figure 3.6)

	VarY <sub>1</sub>	VarY <sub>2</sub>	...	VarY <sub>p</sub>
t <sub>M</sub>	$t_{M1}^1, t_{M1}^2, t_{M1}^3$	$t_{M2}^1, t_{M2}^2, t_{M2}^3$		$t_{Mp}^1, t_{Mp}^2, t_{Mp}^3$

Figure 3.6: Average knots sequence

This knots sequence will be used (w.r.t. each variable) to calculate approximation spline functions.

- Step 5: Calculation of parameters matrix.  
Once we have the B-spline by the mean knots sequence, we will build the matrix containing the control points that is the parameters of the approximation function which will be using during the next analyses.  
Since our purpose is approximating the histogram through a spline function in order to minimize the error that we make in the approximation process, we need another output information, that is a index of goodness of fit. Therefore the index we are take into account will be the minimum value coming from the

### 3.4. Histogram transformation process

---

optimization process. Then, another output will be a matrix containing the relative errors of each histogram.

Another important comment to be done is the following: we started from histogram data that then were normalized in the interval  $[0, 1]$  and we approximated them by means of B-spline function whose control points will be calculated, to get information about the histogram shape. Doing the transformation on  $[0, 1]$  the histogram shape and the spline do not change. (Figure 3.7)

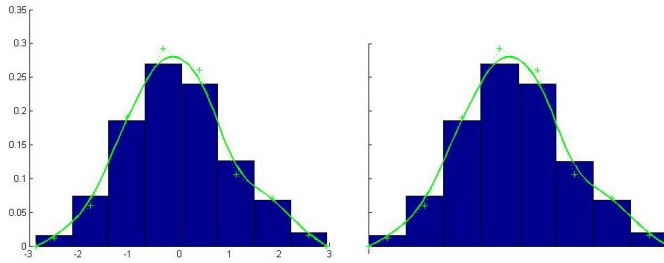


Figure 3.7: Comparing between an histogram and his traslated one

The two splines related to histograms will have the same control points. But since a histogram is a symbolic information that is characterized by three fundamental measures: *location*, *size* e *shape*, we need to retrieve data about *location* and *size*. The information about the histogram *location* come from  $(\max(x) -$

$\min(x))/2$ , while the information about the size come from the width of all the interval that is  $\max(x) - \min(x)$ .

To sum it up, we have a matrix of  $p$  blocks ( $p$  is the number of variables) of order  $m \times 3$  ( $m$  is the number of symbolic units) that represents the information about the *shape*; three matrices of order  $m \times p$  where one gives us information about the *location*, another one about the *size* and the last one about the goodness of fit of the spline to the histograms. Those matrices will be taken into account in the following stages.

# Chapter 4

## Model Data Analysis

After having built the Model Data we propose how to analyze this data. In this chapter we will present two methods: a generalization of Principal Components Analysis in the case of three way matrix called *Multiple Factor Analysis* and a Hierarchical Cluster Analysis based on a definition of a distance between models.

### 4.1 Multiple Factor Analysis

Multiple factor analysis (MFA), proposed by Escofier and Pagès in 1982 [29], studies several groups of variables defined on the same set of individuals. MFA seeks the common structures present in all or some of these sets. The number of variables in each group may differ and the nature of the variables (nominal or quantitative) can vary from one group to the other but the variables should be of the same nature in a given group. The goal of MFA is to integrate different groups of variables describing the same observations. In order to do so, the first step is to make these groups of variables comparable. Such a step is needed because the straightforward analysis obtained by concatenat-

ing all variables would be dominated by the group with the strongest structure.

A similar problem can occur in a non-normalized PCA: without normalization, the structure is dominated by the variables with the largest variance. For PCA, the solution is to normalize each variable by dividing it by its standard deviation.

The solution proposed by MFA is similar: To compare groups of variables, each group is normalized by dividing all its elements by the inverse of the first eigen-value which is the matrix equivalent of the standard deviation. Practically, This step is implemented by performing a PCA on each group of variables. After normalization, the data tables are concatenated into a data table which is submitted to PCA. The data table consisting of a set of individuals ( $I$ ) described by several groups of variables ( $K_j$ ) and each group corresponds to a table  $X_j$  composed of  $v_k$  variables. All the  $X_j$  table are joined to form a single matrix  $X$  as in the figure 4.1.

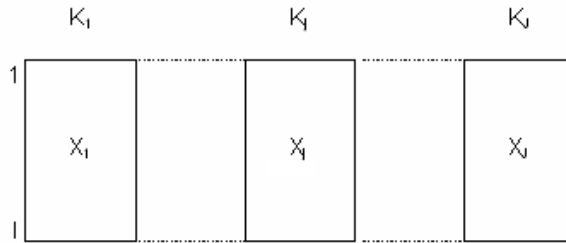


Figure 4.1: Data Matrix  $X$

So, the MFA consists of two steps:

1. Separate Analysis: Each table is analysed separately, that is,  $J$  principal components analysis are performed, that is, one for

#### 4.1. Multiple Factor Analysis

---

each of the  $X_j$  tables. In each case, the first eigen-value for the first factor is selected and denoted by  $\lambda_1^j$

2. Global Analysis: PCA is run on table  $X$ , which is formed by juxtaposing the  $J$  table  $X_j$ . In this analysis each of the  $X_j$  tables is weighted by the inverse of the first eigen-value given by the PCA of the proper table,  $1/\lambda_1^j$

The individual data sets are then projected onto the global analysis to analyze communalities and discrepancies.

All elements (individuals, variables, groups) are represented in Euclidean spaces. These space are named according to the objects they include:

$R^K$  individuals space (defined by all the variables),

$R_j^K$  individuals space defined by group  $j$  variables,

$R^I$  variables space,

As said be-

$E_j$  sub-space of  $R^I$  spanned by variables of group  $j$ ,

$R^{I^2}$  groups space.

fore, the problem can be roughly decomposed through three items, each one corresponding to a point of view:

- (a) typology of individuals described by the whole set of variables,
- (b) overview of relationships between variables,
- (c) comparison of variables groups.

Items (a) and (b) are classic in factor analysis: Principal Components Analysis; Multiple Correspondence Analysis. Item (c) overlaps several objectives described afterwards.

Whatever the point of view, weighting of variables groups is necessary to make the influence of group comparable in a global analysis. Concretely, according to the factor analysis point of view, we want to avoid the possibility of a single group having a dominant influence on

the first factor (nothing can be required for further factors because a multidimensional group will always influence more factors than an unidimensional one).

The weighting of groups brings about the first eigen-value of factor analysis applied to the single group  $j$  became 1. Therefore, groups are balanced in the following sense: in any direction, maximum inertia of the sub-cloud associated to one group is 1. Thus, in a global factor analysis, it is impossible for a single group to give rise to the first factor.

Of course, group contributions to global analysis are not similar: an unidimensional group cannot exert an important influence on more than one factor; a multidimensional group will influence several factors.

The representation of individuals and variables correspond to the classic aim of factor analysis, that is to say:

- typology of individuals,
- typology of variables,
- links between the two typologies.

Each group defines a structure on the individuals set. A structure defined by group  $j$  is expressed by the shape of cloud  $N_I^j$  which represents an individuals set in PCA of  $X_j$  ( $N_I^j$  belongs to  $R^{K_j}$ ).

In order to compare clouds  $N_I^j$  one to another, we need a superposed representation of  $N_I^j$  which sets up the structure common to the different clouds.

In order to get a global comparison of groups, we need a display in which each group is represented by one point. In a space  $I \times I$  dimension, MFA searches a sequence of dimensions such as each one:

- is associated to a single direction of the variables, space  $R^I$ . That constraint necessarily reduces the goodness of fit but ensures the interpretability of the dimensions.

- maximizes, with usual orthogonality conditions, sum of projections (and not sum of squares). It possesses the disadvantage, very un-



pleasant from a theoretical point of view, of being satisfied only by dimensions and not by sub-space. But this disadvantage is the price to pay in order to obtain properties, analogous to duality relationships in factor analysis, which ensure coherence with previous points of view. Those properties are the following: - the  $s$ -order axe found in the space  $R^{I^2}$  is the scalar product matrix associated to the  $s$ -order principal component of  $X$  found in  $R^I$ . Hence, these directions have the same interpretation;

- the coordinate of  $W_j D$  (where  $W_j$  is the scalar products matrix associated to group  $j$  and  $D$  is the individuals weights matrix) with respect to the  $s$ -order axe (in  $R^{I^2}$ ) is equal to projected inertia of group  $j$  variables along the direction defined by the  $s$ -order principal component in  $R^I$ . A proximity between two groups along direction  $s$  indicates that the common factor  $s$  has the same importance in the two groups.

## 4.2 Clustering analysis for Model Data

As in chapter 2 we had to define distance measures for interval data and histogram data, now we have to define a distance measure for Model Data. The idea is to use the Inter-Models distance proposed by Lauro, Romano, Giordano (2006) and generalize it to this particular kind of data.

The distance introduced in [52] is based on a linear combination of two distances embedding information both on the estimated parameters and on the model fitting.

### 4.2.1 The Inter-Model Distance

Lauro, Romano, Giordano introduce a Inter-Model (IM) distance which is able to take into account both the analytical structure of the models -through the difference between the estimated parameters- and the in-

formation about the model fitting through the difference between the adjusted  $Adj - R^2$  indexes related to each pair of models.

They consider a collection of utility models  $M = \{m^1, \dots, m^j, \dots, m^J\}$ , where each entity  $m^j$  is a  $K$ -dimensional vectors defined as:

$$m^j = (w^j_1, \dots, w^j_k, \dots, w^j_K) \quad (4.1)$$

where the values of  $w^j_k$  is the information related to the  $j - th$  model. The first  $(K - 1)$  values are the estimated model parameters, the  $K$ -th value is the information related to the model fitting. For each of the  $J$  fitted utility models the part-worth coefficients  $(w^j_1, \dots, w^j_k, \dots, w^j_{K-1})$  are assumed to be estimated by Ordinary Least Square (*OLS*). They propose to use a statistical index of model fitting (i.e. the  $Adj - R^2$ ) as supplementary information about the utility functions in order to exploit the actual predictive power of the utility model.

Thus, let  $M$  the data collection (table 4.1) it consists of two kinds of information: the analytical terms and the statistical model fitting.

Utility models	Analytical functional form	Statistical model fitting
<i>Model 1</i>	$w^1_1, \dots, w^1_k, \dots, w^1_{K-1}$	$w^1_K$
$\dots$	$\dots$	$\dots$
<i>Model j</i>	$w^j_1, \dots, w^j_k, \dots, w^j_{K-1}$	$w^j_K$
$\dots$	$\dots$	$\dots$
<i>Model J</i>	$w^J_1, \dots, w^J_k, \dots, w^J_{K-1}$	$w^J_K$

Table 4.1: The data collection

The two pieces of information are combined to define the following measure:

$$IM(m^j, m^{j'}|\lambda) = \lambda IM_p + (1 - \lambda) IM_r \quad (4.2)$$

with  $\lambda \in [0, 1]$ . The  $IM$  measure is a convex combination of two quantities  $IM_p$  and  $IM_r$ , where  $IM_p$  is the  $L_2$ -norm between the estimated

parameters:

$$IM_p = \left[ \sum_{k=1}^{K-1} \left( w_k^j - w_k^{j'} \right)^2 \right]^{\frac{1}{2}} \quad (j \neq j') \quad (4.3)$$

and  $IM_r$  is the  $L_1$ -norm between the  $Adj - R^2$ :

$$IM_r = \left| w_K^j - w_K^{j'} \right| \quad (j \neq j'). \quad (4.4)$$

Let us consider  $J$  models  $m^j$  from an arbitrary input space  $\Omega$ , the function

$$MD(m^j, m^{j'} | \lambda) : \Omega \times \Omega \rightarrow R^+$$

satisfies the following conditions:

1.  $IM(m^j, m^{j'} | \lambda) \geq 0$  and  $IM(m^j, m^{j'} | \lambda) = 0 \forall m^j = m^{j'} \in \Omega$
2.  $IM(m^j, m^{j'} | \lambda)$  is symmetric, i.e.  $IM(m^j, m^{j'} | \lambda) = MD(m^{j'}, m^j | \lambda)$
3.  $IM(m^j, m^{j'} | \lambda) \leq IM(m^j, m^{j^*} | \lambda) + IM(m^{j^*}, m^{j'} | \lambda) \forall m^j, m^{j'}, m^{j^*} \in \Omega$

Propositions 1) 2) are direct consequence of the  $IM$  definition as a convex combination of the two euclidean distances. The 3) can be shown as follows:

For the first adding term we have for each  $j \neq j' \neq j^*$ :

$$\begin{aligned} & \lambda \left[ \sum_{k=1}^{K-1} \left( w_k^j - w_k^{j'} \right)^2 \right]^{\frac{1}{2}} \leq \\ & \lambda \left\{ \left[ \sum_{k=1}^{K-1} \left( w_k^j - w_k^{j^*} \right)^2 \right]^{\frac{1}{2}} + \left[ \sum_{k=1}^{K-1} \left( w_k^{j^*} - w_k^{j'} \right)^2 \right]^{\frac{1}{2}} \right\} \end{aligned} \quad (4.5)$$

while, for the second adding term we have:

$$(1 - \lambda) \left| w_K^j - w_K^{j'} \right| \leq (1 - \lambda) \left\{ \left| w_K^j - w_K^{j^*} \right| + \left| w_K^{j^*} - w_K^{j'} \right| \right\} \quad (4.6)$$

then

$$\begin{aligned}
 ID(m^j, m^{j'}|\lambda) \leq & \lambda [ID_p(m^j, m^{j*}|\lambda) + ID_p(m^{j*}, m^{j'}|\lambda)] + \\
 & + (1 - \lambda) [ID_r(m^j, m^{j*}|\lambda) + ID_r(m^{j*}, m^{j'}|\lambda)]
 \end{aligned}
 \tag{4.7}$$

that is

$$IM(m^j, m^{j'}|\lambda) \leq IM(m^j, m^{j*}|\lambda) + MD(m^{j*}, m^{j'}|\lambda).$$

It follows that the defined function  $IM(m^j, m^{j'}|\lambda)$  is a distance. The value of  $\lambda$  plays the role of a merging weight of the two components  $IM_p$  and  $IM_r$ . In the trivial case when  $\lambda = 1$  the distance  $IM(m^j, m^{j'}|\lambda)$  is defined as a function of the coefficients. We look for a  $\lambda$ -value for the set of models, taking the explicative power of the theoretical models into account.

The definition of the model distance  $IM$  takes the explicative power of each pair of models into account, so that two models with similar estimated coefficients are differentiated for their fitting values. Of course, if two models have different coefficient values they should not be moved closer because of a similar fitting measure. For this reason, the trimmer value of  $\lambda$  should not be less than a given level.

### 4.2.2 Clustering utility functions

To classify the  $J$  utility functions we need to use an unsupervised clustering technique on these  $K$  models parameters. Classical method as hierarchical classification can be suitably used for this purpose. In the following we describe the hierarchical classification method which is particularly well adapted here.

Our proposal for performing a hierarchical segmentation strategy differs with respect to the way in which the similarity between two respondents is determined. In the definition of the distance  $MD$  the

value of  $\lambda$  has not been univocally identified. Indeed, the choice of  $\lambda$  is determined contextually by the classification phase.

The proposed strategy consists in replicating the classification phase and in computing the related cophenetic coefficient on a finite grid of  $\lambda \in [\frac{1}{K-1}, 1 - \frac{1}{K-1}]$ . The selected  $\lambda^*$  minimizes the distortion measure of the classification. At the optimum the value of  $\lambda$  have to maximize the Cophenetic Correlation Coefficient (*Coph*) defined as:

$$Coph(m^j, m^{j'} | \lambda) = \frac{\sum_{m^j < m^{j'}} (MD_{m^j, m^{j'}} - \overline{MD}) (\widetilde{MD}_{m^j, m^{j'}} - \overline{\widetilde{MD}})}{\left[ \sum_{m^j < m^{j'}} (MD_{m^j, m^{j'}} - \overline{MD})^2 \sum_{m^j < m^{j'}} (\widetilde{MD}_{m^j, m^{j'}} - \overline{\widetilde{MD}})^2 \right]^{\frac{1}{2}}} \quad (4.8)$$

where  $MD_{m^j, m^{j'}}$  is the distance between each pairs of rows in the matrix  $M_{(J,K)}$  and  $\widetilde{MD}_{m^j, m^{j'}}$  corresponds to the linkage distances between the objects paired in the clusters.

This coefficient measures the distortion of the classification, indicating how the data fits into the tree-structure obtained.

Given a set  $J$  of model  $\{m^1, \dots, m^j, \dots, m^J\}$ , the steps of the algorithm, on which the strategy is based, can be summarized as follows:

1. *initialization phase*: the computation of  $MD_p$  and  $MD_r$  is carried out, and a grid of  $\lambda \in [\frac{1}{K-1}, 1 - \frac{1}{K-1}]$  is settled with a user defined granularity;
2. *classification phase*:  $\forall \lambda$  the MD are computed, and according to an aggregating criterion (e.g. Ward) the clustering algorithm is performed. The linkage distance relating the tree structure (dendrogram), about each pairs of objects, is retrieved;
3. *optimization phase*: for all cluster structures the cophenetic coefficient is obtained, and the value of  $\lambda^*$  is chosen so that the cophenetic coefficient is maximum;

The final partition is settled according to the distance:

$$MD(m^j, m^{j'} | \lambda^*) = \lambda^* MD_p + (1 - \lambda^*) MD_r \quad (4.9)$$

with  $\lambda^* \in \left[\frac{1}{K-1}, 1 - \frac{1}{K-1}\right]$ .

The outcome of a hierarchical classification strongly depends on the choice of between-individuals and between-clusters distance. Choosing the criterion of the maximum variation enables us to obtain homogeneous classes, losing between classes heterogeneity. The hierarchical classification is carried out with the Ward agglomerative criteria, which gathers, at each step, the closest clusters. The number of clusters is chosen by visually inspecting the hierarchical tree structure.

### 4.2.3 Distance between Model Data

In the case of Model Data we have  $p$  block matrix where each one can be seen as a data collection  $M$  presented in the 4.2.1; the coefficient are the control point of the b-spline and the model fitting is the index of goodness of fit ( $I$ ) proposed in the chapter 3. Besides, we have another two pieces of information on the location and the size indicated respectively with  $a$  and  $b$ . So our idea is to reformulate the Inter-Model distance as the sum of 3 components:

1. a convex linear combination of two quantities, the control points and the error term;
2. a  $L_1$  distance between the location terms;
3. a  $L_1$  distance between the size terms.

Practically we have the following distance:  $\forall j \neq j'$

$$\lambda \left[ \sum_{k=0}^K (p_{jk} - p_{j'k})^2 \right]^{\frac{1}{2}} + (1 - \lambda) |\epsilon_j - \epsilon_{j'}| + |a_j - a_{j'}| + |b_j - b_{j'}| \quad (4.10)$$

We can show that 4.10 is yet a distance:

The new proposal is to generalize this distance in the case of block matrix in the following way:

$$\sum_{h=1}^H \left\{ \lambda_h \left[ \sum_{k=0}^K (p_{jkh} - p_{j'kh})^2 \right]^{\frac{1}{2}} + (1 - \lambda_h) |\epsilon_{jh} - \epsilon_{j'h}| + |a_{jh} - a_{j'h}| + |b_{jh} - b_{j'h}| \right\} \quad (4.11)$$

Let us notice that the coefficients  $\lambda_h$  are individually optimized to define the best distance which discriminates among the  $N$  individual models, for each variable.





# Chapter 5

## A case study on real Data

In this chapter it is proposed an application on real data of the methods already described earlier in this work. The goal is to show how the proposed methodology could enhance the classical data analysis based on “punctual” data.

In particular, the case study will allow to better understand how “histogram data” can describe complex phenomena by enriching the information at hand in terms of distributional shapes.

The database under investigation refers to 30 stocks of S&P MIB observed from 2004 to 2005 regarding some variables as closing price, opening price, daily minimum and maximum prices, adjusted closing price, and volume.

All the methodologies have been implemented through peculiar routines created in Matlab able to calculate the Model Data and the Cluster Analysis, while for the Multiple Factor Analysis the Xl-stat packet has been used.

## 5.1 Data Structure

The available data have been organized in a matrix of the dimension 6000 (200 days for 30 stocks) times 6 variables. The first step is to create new variables, which are typical for financial and for technical analysis, from this initial information: the return calculated as the difference between two subsequent days of closing prices and the volatility of five days calculated as a standard deviation of the return. Notice that to construct these new variables, five observations for each stock are loss.

Consequently, some other variables have been computed: the “mean open-close” and the “size open close” calculated as the average and the absolute deviation between the open and close variables. The same reasoning has been applied by computing the “mean low-high” variable and the “size low-high” one.

Therefore, the whole set of variables considered in the analysis is: “mean open-close”, “size open-close”, “mean low-high”, “size low-high”, “volumes”, “adj close”, “returns”, “volatility”.

At first glance, the observed data matrix leads us to the elimination of the “seat pagine gialle” stock because it shows no variation and it is useless to our aims.

The data matrix will therefore be structured as in the figure 5.1.

The first step is to compute the histogram data for each variable and to build the histogram matrix as illustrated in figure 3.3.

Thus, each histogram in this matrix is approximated by a B-spline functions according to the algorithm illustrated in the third chapter. It brings back to the model matrix (see figure 5.2) synthesized through the parameters so obtained.

Therefore, we work on the block matrix (see figure 1.3) where each block is formed by the model parameters which now become variables describing all stocks. (See figure 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10).

## 5.1. Data Structure

Date	Mean Open Close	Size Open-Close	Mean Low Size	Size Low High	Volume	Adj Close*	Returns	Volatility	Titolo
06-lug-04	16,63	-0,14	16,57	0,32	22923700	16,7	0,19	0,175367	Eni
07-lug-04	16,66	0,2	16,655	0,21	17062300	16,56	-0,14	0,143771	Eni
08-lug-04	16,695	-0,01	16,595	0,27	17751100	16,7	0,14	0,136308	Eni
07-apr-05	20,57	-0,46	20,55	0,5	24983000	20,8	0,51	0,234243	Eni
08-apr-05	20,645	0,27	20,605	0,35	24640700	20,51	-0,29	0,290775	Eni
11-apr-05	20,36	-0,04	20,325	0,23	15060900	20,38	-0,13	0,300915	Eni
06-lug-04	4,63	0,02	4,615	0,05	69627400	4,62	-0,01	0,007071	Tim
07-lug-04	4,62	0	4,635	0,03	35086600	4,62	0	0,006937	Tim
08-lug-04	4,615	-0,01	4,605	0,03	32574800	4,62	0	0,005477	Tim
07-apr-05	5,19	-0,1	5,2	0,12	40038300	5,24	0,1	0,049295	Tim
08-apr-05	5,245	0,07	5,235	0,11	23324700	5,21	-0,03	0,051284	Tim
11-apr-05	5,2	-0,04	5,22	0,08	92604000	5,22	0,01	0,047958	Tim
06-lug-04	4,04	0,06	4,04	0,08	79973800	4,01	-0,06	0,029665	Unicredit
07-lug-04	4	0,02	3,995	0,05	59337800	3,99	-0,02	0,02881	Unicredit
08-lug-04	3,995	-0,03	3,99	0,04	29829500	4,01	0,02	0,031145	Unicredit
07-apr-05	4,57	0	4,57	0,04	22705300	4,57	0	0,034205	Unicredit
08-apr-05	4,59	0	4,59	0,04	21084900	4,59	0,02	0,032863	Unicredit
11-apr-05	4,59	0	4,605	0,03	16348900	4,59	0	0,024495	Unicredit

Figure 5.1: Financial Data Matrix

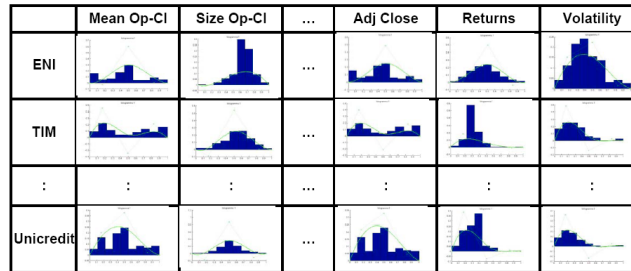


Figure 5.2: Histogram Matrix approximated by b-spline

Titoli	Mean Open-Close					
	Mean Op-CI1	Mean Op-CI2	Mean Op-CI3	Mean Op-CI err	Mean Op-CI loc	Mean Op-CI amp
Eni	-0,031658	0,60784	0,0070474	0,0012914	18,48	4,33
Tim	0,45223	-0,24388	0,21326	0,00037531	4,89	1,37
:	:	:	:	:	:	:
Unicredit	0,20091	0,37831	0,022219	0,00073169	4,21	0,76

Figure 5.3: Table of “Mean Open-Close” variabile

	Size Open-Close					
Titoli	Size Op-CI1	Size Op-CI2	Size Op-CI3	Size Op-CI err	Size Op-CI loc	Size Op-CI amp
Eni	-0.090009	0.54497	0.0047575	9.68E-05	-4.50E-02	8.30E-01
Tim	-0.011142	-0.0091482	-8.67E-18	0.0015079	-0.04	0.3
:	:	:	:	:	:	:
Unicredito	-0.17265	0.87117	-0.11355	0.00047184	-0.005	0.17

Figure 5.4: Table of “Size Open-Close” variable

	Mean Low-high					
Titoli	Mean L-H1	Mean L-H2	Mean L-H3	Mean L-H err	Mean L-H loc	Mean L-H amp
Eni	-0.015277	0.60703	-0.0087912	0.0011568	18.428	4.355
Tim	0.44887	-0.22902	0.2094	0.00038134	4.89	1.38
:	:	:	:	:	:	:
Unicredito	0.22529	0.38255	0.031421	0.0009093	4.215	0.78

Figure 5.5: Table of “Mean Low-High” variable

	Size Low-high					
Titoli	Size L-H1	Size L-H2	Size L-H3	Size L-H err	Size L-H loc	Size L-H amp
Eni	0.57455	-0.024603	0.0041136	0.00037094	0.31	0.42
Tim	0.42994	-0.080264	0.020705	0.00011725	0.105	0.21
:	:	:	:	:	:	:
Unicredito	0.42067	-0.035027	0.00064289	0.00011175	0.065	0.09

Figure 5.6: Table of “Size Low-High” variable

	Volume					
Titoli	Volume1	Volume2	Volume3	Volume err	Volume loc	Volume amp
Eni	0.82438	-0.50086	0.11955	0.0024739	5.63E+07	1.02E+08
Tim	0.84525	-0.51354	0.12258	0.0023828	1.92E+08	3.84E+08
:	:	:	:	:	:	:
Unicredito	0.58225	-0.16752	0.059333	0.00013486	6.70E+07	1.15E+08

Figure 5.7: Table of “Volume” variable

## 5.2 Case study: Multiple Factor Analysis

In this case study it is supposed that financial analysts wish to use the data in order to describe the different assets and select a suitable

## 5.2. Case study: Multiple Factor Analysis

---

Titoli	Adj Close*					
	Adj Close*1	Adj Close*2	Adj Close*3	Adj Close*err	Adj Close*loc	Adj Close*amp
Eni	0,00033978	0,488	0,02932	0,00078504	18,55	4,5
Tim	0,4002	-0,21399	0,18843	0,00035224	4,89	1,38
:	:	:	:	:	:	:
Unicredito	0,21785	0,32868	0,045128	0,00057655	4,22	0,78

Figure 5.8: Table of “Adj Close” variable

Titoli	Returns					
	Returns1	Returns2	Returns3	Returns err	Returns loc	Returns amp
Eni	0,0065066	0,49866	-0,035901	7,60E-05	5,50E-02	9,10E-01
Tim	0,19046	0,027975	-0,0080651	0,0028551	0,085	0,43
:	:	:	:	:	:	:
Unicredito	0,38697	-0,07374	0,0099555	0,00075608	0,045	0,23

Figure 5.9: Table of “Returns” variable

Titoli	Volatility					
	Volatility1	Volatility2	Volatility3	Volatility err	Volatility loc	Volatility amp
Eni	0,15001	0,22404	0,013273	0,00015922	0,18236	0,27824
Tim	0,48487	-0,13894	0,025238	6,48E-05	6,89E-02	1,38E-01
:	:	:	:	:	:	:
Unicredito	0,62401	-0,2356	0,055573	3,94E-05	4,82E-02	8,22E-02

Figure 5.10: Table of “Volatility” variable

portfolio according to some desirable features of the stocks. Indeed, it is essential to describe the whole set of the S&P MIB shares according to their own structural characteristics. Let us note that such features has been captured by the model parameters coding since they take into account valuable information about location (level) and size (variability, i.e. risk) for each asset.

Aiming at exploring such characteristics we exploit the powerful tool of graphical display and geometrical interpretation given by multidimensional data analysis.

According to the three-way data structure a Multiple Factor Analysis

is carried out. The analysis is performed by considering 8 matrices (one for each variable). Each matrix is 29 times 6, where the six characteristic parameters of the models are in columns: the first three parameters are the three B-spline control points, the fourth parameter is the error term, the fifth and the sixth are the location and the width of the histogram.

The first phase of the AFM consists of eight separate principal component analysis, one for each matrix. In the following some results are showed.

The analysis executed on the first block regards the “mean open-close” variable; the most significant factorial axis are the first two that are able to explain 68,41% of the variability (figure 5.11).

	F1	F2	F3	F4	F5	F6
Autovalore	2,358	1,746	0,952	0,770	0,130	0,043
Variabilità (%)	39,302	29,104	15,865	12,839	2,168	0,723
% cumulata	39,302	68,405	84,270	97,109	99,277	100,000

Figure 5.11: Eigenvalue of first matrix

Studying the correlations between the variables and the factorial axis it is evident that the variables characterizing the first factorial axis are “mean open-close error”, “mean open-close location” and “mean open-close width”, while the second factorial axis is characterized by the “mean open-close1”, “mean open-close2”, and “mean open-close3” variables.

Therefore we are able to establish that the first factorial axis gives us information on the size of the histogram while the second one gives information on the shape (see figure 5.13).

The combined graphic of observation and variables allows us to

## 5.2. Case study: Multiple Factor Analysis

	F1	F2	F3	F4	F5	F6
Mean Open-Close1	-0,533	-0,425	0,710	-0,005	0,176	0,018
Mean Open-Close2	0,561	0,769	-0,030	-0,190	0,230	0,056
Mean Open-Close3	-0,317	-0,680	-0,639	-0,008	0,160	0,053
Mean Open-Close err	0,639	-0,143	0,026	0,750	0,082	-0,037
Mean Open-Close loc	0,753	-0,500	0,023	-0,404	0,061	-0,128
Mean Open-Close amp	0,827	-0,492	0,192	-0,089	-0,100	0,139

Figure 5.12: Correlation between variables and factor of the first matrix

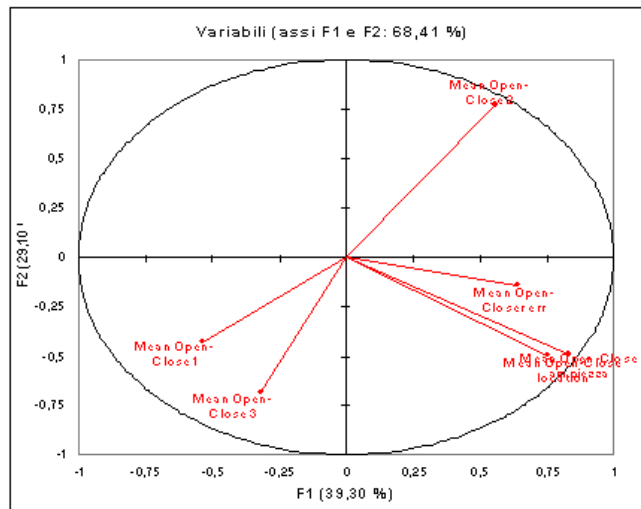


Figure 5.13: Correlation Circle of the first PCA

decide which stocks are characterized by more or less high shape and size values.

You can clearly see (figure 5.14), for example, that the Banca Antonveneta is characterized by high location and size values of the “mean open-close” variable. Furthermore the first axis is also characterized by the error parameter which allows us to stabilize for what

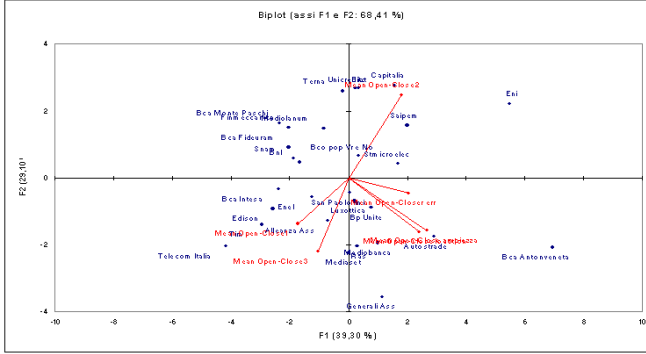


Figure 5.14: Joint graphic of observation and variables (first PCA)

stock we have obtained the better or worse approximation.

In the analysis of the “size open-close” variable, it results that the first two factorial axis explain 68,99% of the variability and that the first factorial axis is characterized by the 1st, 2nd, and 3rd parameters that give us information on the histogram shape while the second axis is characterized by location and width, i.e. information on size.

In the correlations matrix between variables and factors (figure 5.16)

	F1	F2	F3	F4	F5	F6
Autovalore	2,738	1,402	0,872	0,837	0,110	0,042
Variabilità (%)	45,628	23,362	14,535	13,944	1,826	0,705
% cumulata	45,628	68,990	83,525	97,469	99,295	100,000

Figure 5.15: Eigenvalue of second matrix

is shown that the error parameter is strongly correlated to the third axis.

In the joint graphic of observations and variables (figure 5.18) is clear, for example, that Banca Antonveneta is also characterized by



## 5.2. Case study: Multiple Factor Analysis

	F1	F2	F3	F4	F5	F6
Size Open-Close1	-0,942	0,227	-0,050	-0,133	-0,159	0,128
Size Open-Close2	0,962	0,175	0,102	-0,028	0,102	0,150
Size Open-Close3	-0,769	-0,480	-0,264	0,246	0,213	0,048
Size Open-Close error	-0,447	0,234	0,809	0,294	0,060	-0,004
Size Open-Close locatio	-0,351	0,801	-0,136	-0,437	0,155	-0,034
Size Open-Close ampiez	0,099	0,627	-0,341	0,693	-0,029	-0,006

Figure 5.16: Correlation between variables and factor of the second matrix

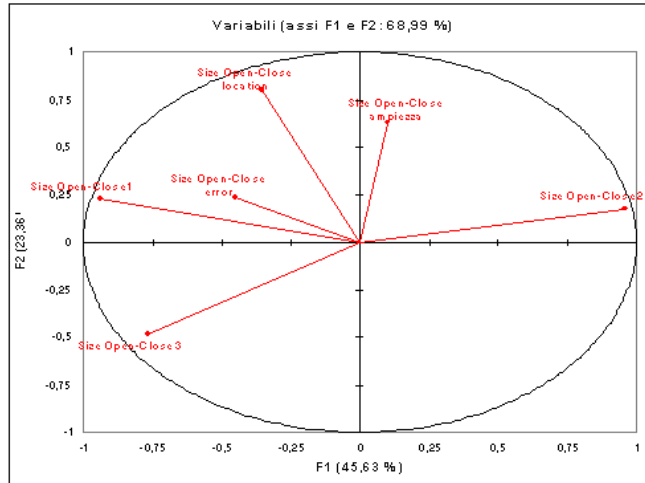


Figure 5.17: Correlation Circle of the second PCA

high values in the location and width parameters of the “size open-close” variable.

Similar results can be achieved when considering the analysis of the other variables.



## 5.2. Case study: Multiple Factor Analysis

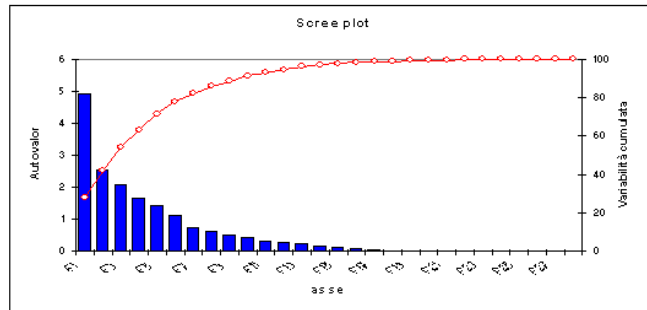


Figure 5.19: Scree plot

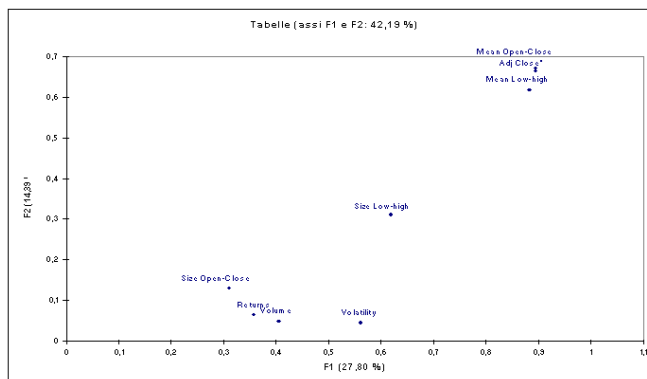


Figure 5.20: Map of tables

the two groups of variables. It is valuable the proximity between the variables representing the prices like “mean open-close”, “mean low-high” and “adj-vol”; a proximity between the variables that expresses variability as “size open-close”, “volatility” and “returns”.

Interesting output is achieved from the correlation circle of the partial axis (see figure 5.22) and from the mapping of the observations (see

	Mean Op-Cl	Size Op-Cl	Mean L-H	Size L-H	Volume	Adj Close*
Mean Open-Close	1,000	0,211	0,975	0,358	0,191	0,913
Size Open-Close	0,211	1,000	0,213	0,385	0,121	0,224
Mean Low-high	0,975	0,213	1,000	0,370	0,210	0,898
Size Low-high	0,358	0,385	0,370	1,000	0,159	0,379
Volume	0,191	0,121	0,210	0,159	1,000	0,205
Adj Close*	0,913	0,224	0,898	0,379	0,205	1,000
Returns	0,247	0,438	0,262	0,324	0,161	0,252
Volatility	0,303	0,173	0,301	0,293	0,338	0,295
AFM	0,826	0,522	0,831	0,629	0,436	0,819

Figure 5.21: RV Coefficients

figure 5.23).

We can see that the first axis represents the risk versus the return,

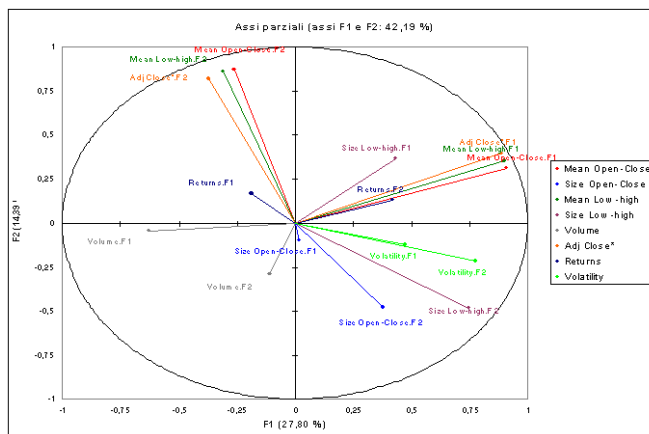
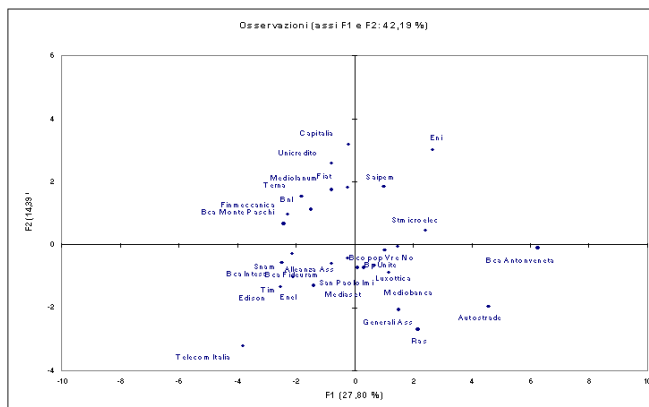


Figure 5.22: Correlation circle of partial axis

and therefore to the left we find stocks with high returns and larger volumes exchanged, while to the right we find stocks with high prices and with a larger risk.

On the other hand the second axis is characterized by stocks that are



### 5.3 Case study: Cluster analysis

It should be noticed that the distance between two individuals in the MFA map is the Euclidean one and all the model parameters (i.e. the variables) share the same importance. The use of a well defined model-distance is able to classify them.

In this section we show the output of the cluster analysis obtained by using the inter-model distance for model data 4.11 proposed in the fourth chapter.

The algorithm produces the tree structure shown in figure 5.24. Tree

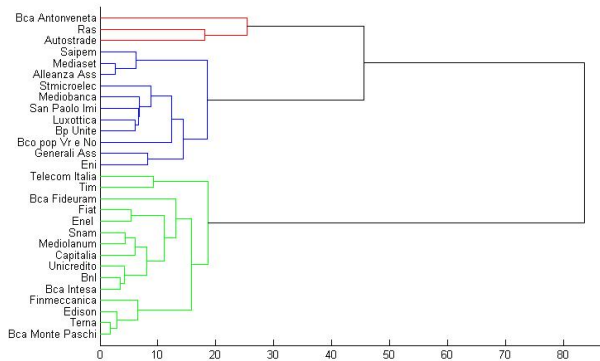


Figure 5.24: The tree structure of the Models

structure shows shows three main groups:

Group 1: Bca Antonveneta, Ras, Autostrade.

Group 2: Telecom Italia, Tim, Bca Fideuram, Fiat, Enel, Snam, Mediolanum, Capitalia, Unicredito, Bnl, Bca Intesa, Finmeccanica,

Edison, Terna, Bca Monte Paschi.

Group 3: Saipem, Mediaset, Alleanza Ass, Stmicroelec, Mediobanca, San Paolo Imi, Luxottica, Bp Unite, Bco pop Vr e No, Generali Ass, Eni.

Indeed some similarity between stocks are expected: groupings clearly defined (banks, insurance companies and utilities/ privatized); Clusters similar to corporate governance as banche popolari (Popolari Unite and Popolare di Verona and Novara); Enel, Snam Edison (Utilities) but also Telecom and Tim (fusion); privatizations (Finmeccanica and Terna).

In general, different factor have contributed on the creation of this clusters. To determine the characteristics of each group, we have built the prototype asset by computing the average of the parametrers in each gruop and then reconstruct the Mean Model. We have the three prototype shown in the figure 5.25. The b-spline, so reconstructed, give us the information on histograms shape for each variable of each group. Furthermore, we need to consider, for each B-spline, the parameter  $\lambda$  (that give us the quality of information), the location parameter and the size one.

In the figure (5.26, 5.27, ??,5.29,5.30,5.31,5.32,5.33) we have plotted, for each variable, the variability against the location. The ideal stocks are located in the lower right, according to a pseudo mean-variance analysis.

As a rule of thumbs, we may decide to choose the best performing cluster and interpret it as an optimal portfolio. Anyway, let us consider that different features of the stocks can lead to different choices. In the case study, the Cluster 1 portfolio seems to be superior according to location characteristics, but looking at the Size characteristics the Cluster 2 looks like a more conservative portfolio (less variability

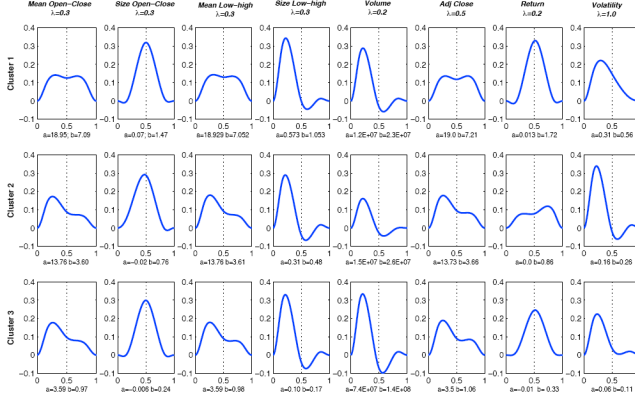


Figure 5.25: Prototypes of the three groups

and so less risk).

In general, in this case study, we refer to a stocks as an object measured through variables represented by models. Each model represents a global evaluation of the risk on each stock. If the objective of a financial operator is to reduce such risk, the reading of the size, locations and shape synthesized in our scheme is valuable. In particular, what is important is certainly the interval among values that one could create in a specific period. Where this interval is particularly high, using the changes in closing prices month by month as a reference, this would mean an increase in volatility on the market. A growth in volatility on the market would clearly mean an increment of the financial risk; an increase of the reduction risk for the finance company. Each single component of portfolio may therefore represent a different risk component if you consider it from this aspect.

In finance we measure these risk models using a specific methodology called VaR value at risk, that is you take single returns (usually of portfolio) and you calculate the expected loss with a definite



### 5.3. Case study: Cluster analysis

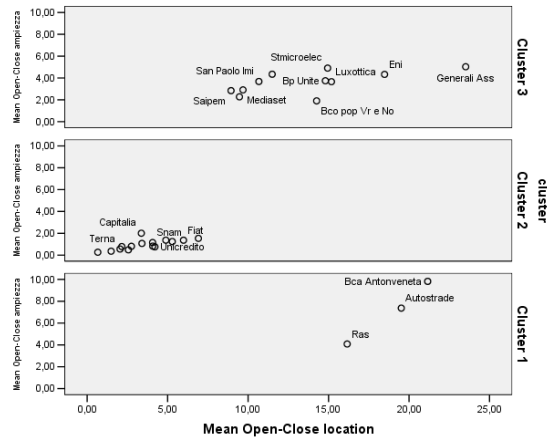


Figure 5.26: Plane variability against location of the “mean open close” variable

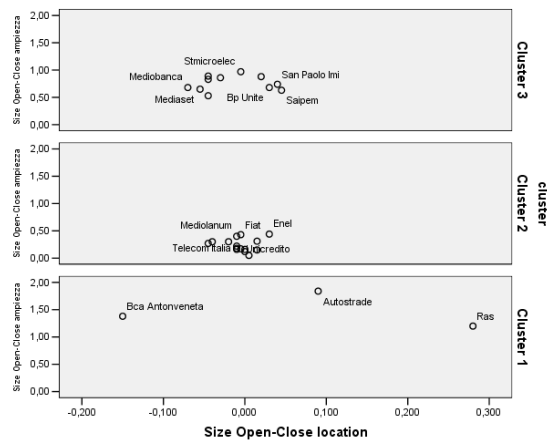


Figure 5.27: Plane variability against location of the “size open close” variable

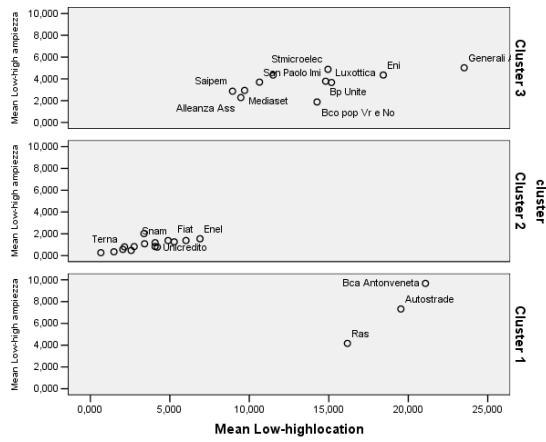


Figure 5.28: output2

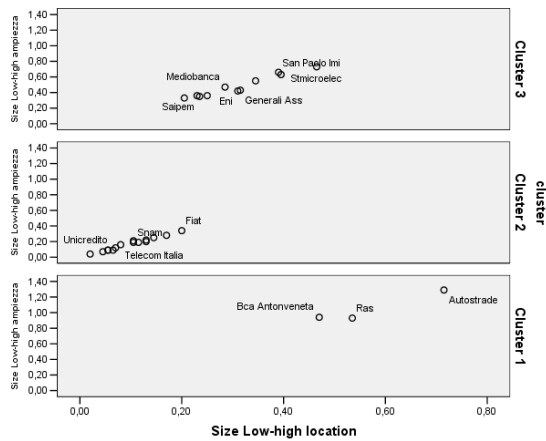


Figure 5.29: Plane variability against location of the “size low high” variable

### 5.3. Case study: Cluster analysis

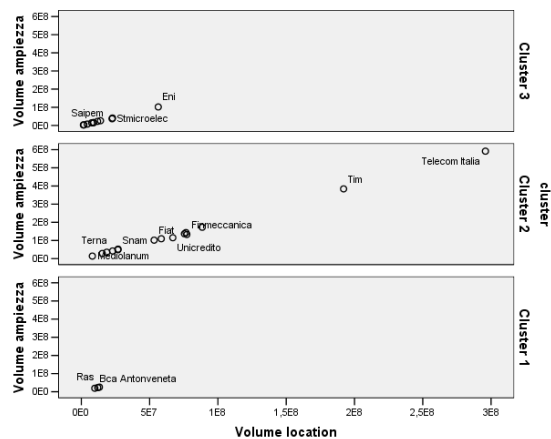


Figure 5.30: Plane variability against location of the “volume” variable

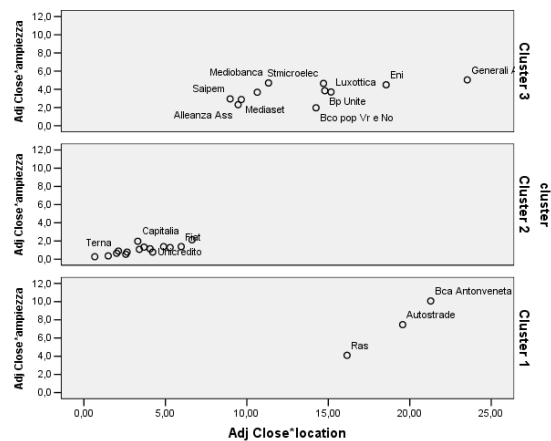


Figure 5.31: Plane variability against location of the “Adj Close” variable

## A CASE STUDY ON REAL DATA

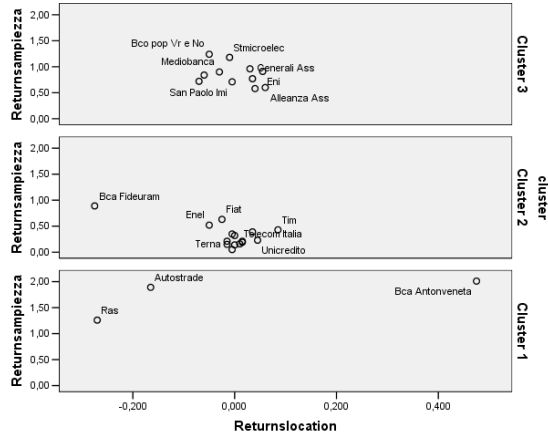


Figure 5.32: Plane variability against location of the “Returns” variable

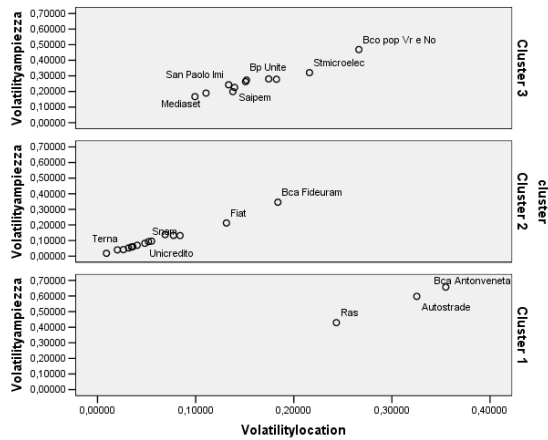


Figure 5.33: Plane variability against location of the “Volatility” variable

confidence interval from a distribution. The hypothesis, that should absolutely be discussed according to many authors, is if the normal used as distribution of base in the VaR is reasonable in these analysis. The answer is usually that the stocks tend not to act in just one way and therefore different hypothesis should be made.

Usually in this literature (for whom has gone further in measuring the quantitative risk from VaR) you use different instruments of statistical kind in measuring the risk. Typically non-parametric instruments as the study of wavelets (in different contexts of study, e.g., of financial time series). In other cases you use non-parametric models to estimate the risk, without any hypothesis, and using non-parametric methods estimating distribution density (and from there you arrive at risk evaluations not based on the hypothesis of the normality).

The models “model” the risk of each possible component subject to symmetric shocks on the system. The groups find models with similar characteristics starting from the shocks of the system and therefore they are important in quantity asset allocation methods since they allow us to visualize the risk in a more consistent form compared to current methods. The groups would therefore be stocks that had the same risk characteristics dinamically speaking where stocks “outside” would have different characteristics (and different risk “forms”).



# Conclusioni

The Symbolic Data Analysis techniques that have been developed during the last decade, represent new and well adoptable instruments for the complex nature of real phenomena. Different methods are developed for different types of data to analyze, which is the starting point for any statistical analysis.

Data of an interval nature can be those concerning a “disciplinary” where each individual is characterized through variables whose values are inside a range. Hence the necessity to analyze phenomena where the data are interval values rather than punctual. However, the interval data give us little information regarding the internal variability while you assume the hypothesis of uniform distribution throughout the interval of phenomenon variation. With the histogram we can obtain this ulterior piece of information about variability in terms of the divergent distribution of the range.

The histogram construction represents a key factor for Histogram Data Analysis. In this thesis we were supposed to have data from big databases and to unite them regarding the different occasions in which a certain number of statistical units have been observed. The ground hypothesis is that histograms thus constructed would have the same number of classes and that they could exceed that same numerosness using the density histogram. The construction of histograms is a problem that is still worked on to determine the optimal number

of classes and their width. This problem is by-passed when working with instruments that allow us to ignore the empirical error.

There are two different types of histograms in literature; the histogram deriving from punctual data, and the histogram deriving from Symbolic Data [6]. This could be interesting if we start with interval variables and we want to unite them in order to build a histogram. For example, in the dataset analyzed in the fifth chapter, if we had considered the interval  $[min, max]$  as a variable of each stock and for each stock we had  $n$  intervals, referred e.g. to  $n$  days, we would have built a histogram based on the overlapping of intervals.

Some characteristics of the histogram are relevant as, e.g. the average of median, the range, the second, third, and fourth moments on which we base the concept of variance, symmetry and kurtosis. The confrontation of these characteristics is fine when we consider normal models or models that are referrable through transformation. A more flexible and immediate approach could be the one based on the confrontation of corresponding classes, where this approach indeed includes the implicit error in every statistical survey in the confrontation.

On these grounds we build the matrix of the histogram data that ensures a histogram which corresponds to every individual and to every variable. Starting from a histogram matrix, we arrive to the problem of analyzing it.

The techniques developed up to the present work on the density of probability or on the cumulated frequencies of the histograms. This thesis want to supply an alternative suggestion regarding histogram analysis. Based on the idea that the *histogram=model+error* we replace each histogram with an approximation function to analyze specific parameters.

The type of function used to approximate the histograms characterizes the choice of the parameters of the model.

In the case of B-splines that are characterized by their capacity to ad-



just themselves to represent rather smooth curves, we have considered the so called control points as parameters that give us information on the histogram form. This form is not referable to a known density function and is defined by parameters not statistically interpretable. In a different context we could consider approximating histograms through density functions of probability as long as they enter in a family of functions and have the same number of parameters and as long as the latter are comparable. The last case brings a certain inflexibility to the model although contributing with a notable simplicity and interpretation. On the other hand the proposed approach results more general considering the former case as a particular case but offers major flexibility in the choice of model and a possible situation comparability not referable to one single theoretical model.

Alternative proposals are the use of moment generating functions or “Lambda Generalized” [22] that can represent a compromise between flexibility and statistical meaning.

In all cases, the objective is, however, to obtain an appropriate transformation of the original histogram data into a series of approximate function parameters for which the data matrix consists in as many parameter blocks as the histogram variables consider.

The methodologies of analysis of this kind of data refer to two types of classical techniques of Multidimensional Analysis; Principal Components Analysis and Cluster Analysis. The former aims at pointing out the structural relationships among the histogram variables, the latter has the intention to recognize similarities and homogeneous groups of symbolic units described by a series of histograms.

In the first case the use of factorial techniques for the study of block matrices, as the Multiple Factorial Analysis proposes, in this context, becomes central.

This does not exclude the possibility to use the proper methods of the three-way-analysis with the advantage to confront the variable parameters transversally one by one, (see figure 5.34).

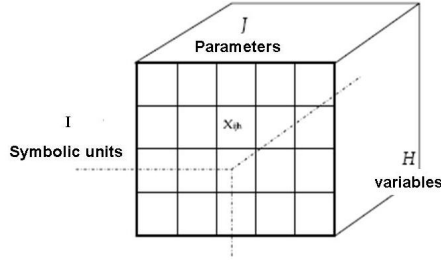


Figure 5.34: Three way array

To classify symbolic models it becomes relevant to define an adequate distance.

The proposed distance (4.11) is the sum of three distances regarding the three characteristics of a symbolic data, the first addend refers to the *shape* the second one to the *location* and the third one to the *size* of the histogram. In this definition not only the comparison of the parameters of each symbolic model becomes relevant, but also their grade of adjustment to the histograms for which the first component is a convex combination of parameters regarding the shape and the approximation error.

In our proposal equal importance has been given to the three addends, but that does not exclude that these weights can be defined by the researcher, based on the problem to analyze. For example, if we were interested in studying a classification based solely on shape we could ignore the other two components.

An ulterior development could be to search for optimal weights that respond to the need of more internally homogeneous symbolic models but maximally different among themselves.

The representation of the symbolic model prototype to describe

each class represents an unsolved problem. In the case of representation through hypercubes, assuming the independence among variables, it furnishes adequate description to the symbolic objects case, described by interval variables, pointing out the three fundamental aspects of these objects, which are location, size, and shape.

In the case of symbolic models relating dimension of the symbolic object there is not a simple geometrical description, if not the respective sets of definitions of the adopted functions.

In figures 5.35, 5.36, 5.37 qualitative representations of symbolic data are shown.

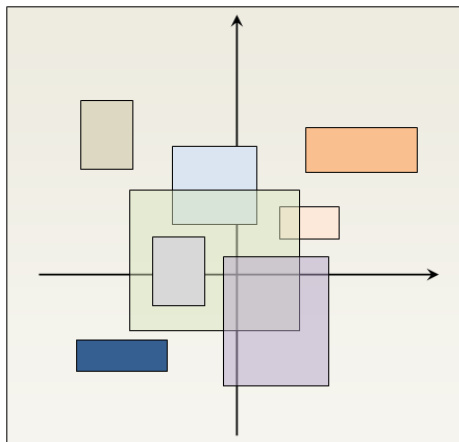


Figure 5.35: Qualitative representation of interval data

In these representations we have the information on interval variance, but not on the covariance between the two objects.

Looking at factorial synthesis, where each factor constitutes an independent variable, and so, in our case, it is in itself a symbolic model; the covariance problem is in fact exceeded.

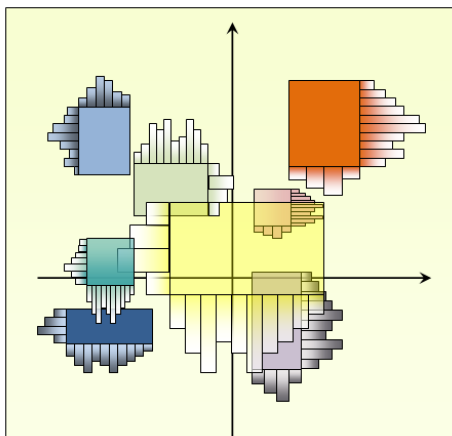


Figure 5.36: Qualitative representation of histogram data

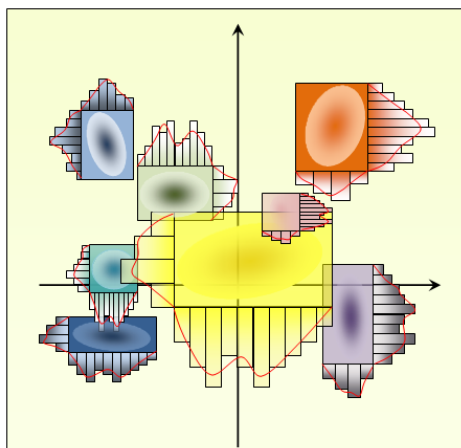


Figure 5.37: Qualitative representation of model data

# Appendix A

## Routine in Matlab Language

In this appendix is reported the Matlab routine. Starting from a matrix in which there are  $p$  variables and  $n$  units observed in  $k$  occasion, the first step is to build histograms pooling the occasions for each unit. In this way we will have a histogram matrix  $X$  of  $n \times p$  order.

Successively, histograms will be replaced by b-spline through a bound-constrained optimization problem. Then Model Data will be built and each histogram will be substituted by a 6-dimensional vector composed of three control points, an error term, and the location and size terms. This new type of data will be classified with the methodology proposed in the fourth chapter.

For the Multiple Factor Analysis the Xl-stat packet has been used.

## A.1 Model Data Building

```
function [ptfin, errfin, location2, size2]=model(X,d)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   MODEL trasforms a thiny data matrix in model data matrices
%
%   INPUT PARAMETERS:
%       X: bidimensional array (nd by p). Columns represent p
%           variables, while rows contain n occasions for each of
%           d symbolic units
%       d: scalar. It is number of symbolic units
%
%   OUTPUT PARAMETERS:
%       ptfin,location2,size2: bidimensional arrays. They contain
%           information about model used for each symbolic units
%       errfin: bidimensional array. Approximation error
%           resulting in substitute symbolic unit by model
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Build 1 by d structure array where each fields (symbolic unit)
% is a n by p array
[n,p]=size(X);
k=n/d;
for i=1:d
    data(i).X=X((i-1)*k+1:i*k,:);
end

for i=1:d %for each unit
    % Compute models that describe data by means of bspline approximation
    % and return model parameters
    [nodi location ampiezza]=model1(data(i).X,0);

    % save model parameters in matrices
    nodi2(i,:)=allin_nodi(nodi);
    location2(i,:)=location';
    size2(i,:)=ampiezza';
end

% compute mean knots sequence
medie_nodi=mean(nodi2);
num=size(medie_nodi,2)/3;
media_nodi=reshape(medie_nodi,3,num)';
for i=1:d %for each unit
    % approximate data by means of bspline built starting from the same
    % knots sequence

```

## A.1. Model Data Building

---

```

    [punticontrollofinali errorefinale]=modelfin(data(i).X, ...
        media_nodi,0);
    % save model parameters in matrices
    ptfm(i,:)=allin_nodi(punticontrollofinali);
    errfin(i,:)=errorefinale';
end

%      data: is a 1 by d structure array where each fields is      %
%              a n by p array; d, n and p are respectively the    %
%              number of different units, occasions and variables %

%-----
function [nodifin location size2]=model1(B,dis)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                    %
%  MODEL1 builds histograms pooling occasions for each symbolic %
%  unit and for each variable. Then histograms are approximate %
%  by B-spline functions.                                         %
%                                                                    %
%  INPUT PARAMETERS:                                              %
%      B: bidimensional array (n by p). Columns represent p      %
%          variables, while rows contain n occasions of one      %
%          symbolic unit                                          %
%      dis: if dis=0 no graphical output are displayed           %
%            if dis=1 graphical output are displayed             %
%                                                                    %
%  OUTPUT PARAMETERS:                                            %
%      nodifin: bidimensional array. ??????????                  %
%      location,size2: bidimensional arrays. They contain        %
%          information about location and size of histogram data %
%                                                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

numosserv=size(B,1);
% Compute number of classes to be used for building histograms with Sturges
% Formula
numclassi=1+ceil(log2(numosserv));
A=B';

for i=1:size(A,1)

    % Standardization of histogram
    Y1=(A(i,:)-min(A(i,:)))/(max(A(i,:))-min(A(i,:)));
    Y(i,:)=Y1;
    location(i,:)=min(A(i,:));
    size2(i,:)=max(A(i,:))-min(A(i,:));

```

```

% Compute histogram
[heights,centers] = hist(Y(i,:),numclassi);
heights=heights/size(Y,2);
Sum=sum(heights)

%Plot histogram if required
if dis==1
    figure(i)
    bar(centers,heights,1);
    hold on
end

%compute bins sequence
n = length(centers);
w = centers(2)-centers(1);
t = linspace(centers(1)-w/2,centers(end)+w/2,n+1);

% set upper and lower bound
for j=1:3
    lb(j)=eps;
    ub(j)=1-eps;
end

%initialize starting point
nodi0=[0.25 0.50 0.75];

% Compute knots sequence that minimize error due to approximation of
% histogram by B-spline
[nodi,fval,flag,output]=fmincon(@(nodi)myfun(nodi,heights,t),nodi0,...
    [],[],[],[],lb,ub,@(nodi)mycon(nodi,t));

nodi_finali=sort(nodi);
nodifin(i,:)=nodi_finali;

%Plot B-spline and its control polygon if required
if dis==1
    Dsp = costruzione(nodi_finali,t,heights);
    diff(nodi_finali);

    % Compute control points (px,py)
    pix=Dsp.knots;
    dimnodi=size(pix,2);
    for k=1:dimnodi-4;
        a(k)=0;
        for j=k+1:k+3;
            a(k)=a(k)+sum(pix(j));
        end
    end
end

```



### A.1. Model Data Building

---

```
        px2=a/3;
    end
    px=px2(2:4);
    py=Dsp.coefs(2:4);
```

```
        fnplt(Dsp,'-g')
        plot(px,py,':g');
    end
end
```

```
%-----
function nodi2=allin_nodi(nodi)

[n,p]=size(nodi);

nodi2=0;

for i=1:n
    nodi2=[nodi2 nodi(i,:)];
end
nodi2(1)=[];
%-----

%-----
% Computing B-spline through an interpolate spline
% with a fix knots sequence.
%
```

```
function Dsp = costruzione(nodi,t,heights)

numt=size(t,2);
numnodi=size(nodi,2);
nodi=sort(nodi);

for k=1:numnodi
    i=1;
    while (nodi(k)>t(i+1))
        i=i+1;
    end
    altezza(k)=heights(i);
end
nodi=[min(t) nodi max(t)];
Ffalsext=[0 altezza 0];

sp=spline(nodi,Ffalsext);
```

## ROUTINE IN MATLAB LANGUAGE

---

```
Dsp=fn2fm(sp,'B-');

%-----
%
function f = myfun(nodi,heights,t)
numt=size(t,2);
Dsp=costruzione(nodi,t,heights);
f=stimaerrore1(Dsp,t,heights,numt);

%-----
function [c,ceq] = mycon(nodi,t1)
nodi=sort(nodi);
nodi=[min(t1) nodi max(t1)];
numnodi=size(nodi,2);
for i=1:numnodi-1
    c(i)=(nodi(i)-nodi(i+1));
    c(i)=c(i)+(t1(2)-t1(1));
    %c(i)=c(i)+eps;
end
%c(numnodi)=abs(nodi(1)-nodi(numnodi))+eps;

%c=[c];

ceq=[];
%-----

%*****
% Computing B-spline starting from mean knots sequence
%*****

function [punticontrollo errore]=modelfin(B,nodimedi,dis)
numosserv=size(B,1);
numclassi=1+ceil(log2(numosserv));

A=B';
for i=1:size(A,1)
    if dis==1
        figure(i)
        hold on
    end
    Y1=(A(i,:)-min(A(i,:)))/(max(A(i,:))-min(A(i,:)));
    Y(i,:)=Y1;

%building histograms
[heights,centers] = hist(Y(i,:),numclassi);
heights=heights/size(Y,2);
```

## A.1. Model Data Building

---

```
%Sum=sum(heights)
%Disegno istogramma 1
if dis==1
bar(centers,heights,1);
end
n = length(centers);
w = centers(2)-centers(1);
t = linspace(centers(1)-w/2,centers(end)+w/2,n+1);

Dsp = costruzione(nodimedi(i,:),t,heights);
%Control Points (px,py)
pix=Dsp.knots;
dimnodi=size(pix,2);

for k=1:dimnodi-4;
    a(k)=0;
    for j=k+1:k+3;
        a(k)=a(k)+sum(pix(j));
    end

    px2=a/3;

end
px=px2(2:4);
py=Dsp.coefs(2:4);
%
Dsp1(i,:)=Dsp;
px1(i,:)=px;
py1(i,:)=py;
t1(i,:)=t;
%Plot B-spline and control polygon.
if dis==1
fnplt(Dsp1(i,:),'-g')
plot(px1(i,:),py1(i,:),':g',px1(i,:),py1(i,:),'+g');
title('Istogramma 1');
end

e=stimaerrore1(Dsp1(i,:),t1(i,:),heights,numclassi+1);
err=e/(max(t1(i,:))-min(t1(i,:)));
errore(i,:)=err;
end

nodi=px1;
punticontrollo=py1;%-----
```

## A.2 Cluster Analysis on Model Data

```

%*****
% X is the control points matrix, E is the error matrix, A is the location
% matrix and B is the size matrix.
%
% Il programma suddivide prima tutta la matrice in tante matrici quante
% sono le variabili considerando che per ogni variabile abbiamo determinato
% 3 punti di controllo. Effettua la cluster per ogni matrice definendo un
% lambda ottimale per ognuna di esse ed infine effettua una nuova cluster
% considerando la somma delle distanze di tutti i blocchi di matrici di
% variabili ognuno con il rispettivo lambda determinato in maniera ottimale
%*****

```

```

function [matriceparametri Z]=divisioneparametri(X,E,A,B,k,legame)
if nargin == 5
    legame = 'ward';
end;

%% k è il numero di colonne di ogni matrice da ottenere(numero di punti di controllo per ogni variabile=3
%% s=0 non cancellare riga k+1, s=1 cancella la riga k+1

[n,p]=size(X);

d=p/k;

for i=1:d;
    matriceparametri(i).X=X(:,1:k);
    X(:,1:k)=[];
    matriceparametri(i).E=E(:,1);
    E(:,1)=[];
    matriceparametri(i).A=A(:,1);
    A(:,1)=[];
    matriceparametri(i).B=B(:,1);
    B(:,1)=[];
end
for i=1:d

    [Amax, C, VD]=classnew(matriceparametri(i).X,matriceparametri(i).E,matriceparametri(i).A,matriceparametri(i).B);
    matriceparametri(i).Amax=Amax;
    matriceparametri(i).C=C;
    matriceparametri(i).VD=VD;
end
t=(n/2)*(n-1);
VDTOT=zeros(1,t);
for i=1:d;

```

## A.2. Cluster Analysis on Model Data

---

```
VDTOT=VDTOT+matriceparametri(i).VD;
end
VDTOT;
Z=linkage(VDTOT,legame);

% %t=altezza del taglio
I = inconsistent(Z); % I = criterio per scegliere il numero delle classi
[MI,alt]=max((I(:,4)));
kk=size(Z,1);
t=( Z(alt,3))+ (Z((alt-1),3)) )/2;
%t=Z(alt,3)-0.1
%modificato il 14 Marzo a scopo illustrativo
figure;
[H,N]=dendrogram(Z,0,'colorthreshold',400,'ORIENTATION','right');
%[H,N]=dendrogram(Z);
% Disegna la linea
% X=1:(size(Z,1)+1);
% line(X,t,'LineWidth',2);
% %
% G=cluster(Z,'Cutoff',t)
% G = cluster(Z,'MAXCLUST',3)
%-----
function [Amax, C,VD] = classinew(PF,E,A,B,legame);

warning off
if nargin == 4
    legame = 'ward';
end;
d=size(E,2);

i=1;

for alfa=1:-0.1:0.2;
    VD=misure(PF,E,A,B,alfa);
    Z = linkage(VD,legame);
    C(i,:)=[cophenet(Z,VD),alfa]; %criterio da max per individuare la migliore classificazione
    i=i+1;
end

% for i=1:d
%     figure(i)
%     hold on;
%
% plot(C(:,2),C(:,1));
% end
```

## ROUTINE IN MATLAB LANGUAGE

---

```

[CophMax,Id]=max(C(:,1));
MAXC=C(Id,:);
Amax=C(Id,2);
VD=misure(PF,E,A,B,Amax);
%VD1=VD';

C=C(:,1);
C=C';
%-----
%////////////////////////////////////////////////////////////////////
% Questa funzione costruisce il vettore delle distanze di lunghezza %
% k*(k-1)/2                                                                %
% Parametri input:                                                            %
% PF = Parametri delle funzioni nell'ordine (R1, R2, f1, f2)                %
% dove f# = (a, b1, b2,...)                                                  %
% Parametri output:                                                            %
% VD = Vettore delle distanze                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [VD] = misure(PF,E,A,B,alfa)
k=size(PF,1);
%Dist=zeros(k,k);
p=size(PF,2);
%f1=PF(1,2:p);
%ciclo per l'impilazione del vettore

c=0;
for i = 1:k-1;
    for j = i+1:k;
        c=c+1;
        dpar=(PF(i,:)-PF(j,:));
        d1(c)=(sqrt(sum(dpar.^2)));
    end;
end;
d1=d1';
sd=max(d1(:,1))- min(d1(:,1));
c=0;
for i = 1:k-1;
    for j = i+1:k;
        c=c+1;
        p=d1(c);
        sd=sd;
        vd(c)= epcq(E(i), E(j),A(i),A(j),B(i),B(j), alfa, sd, p);
    end;
end;
%m=max(vd);
%z = 0:0.1:m;
%figure;

```

## A.2. Cluster Analysis on Model Data

---

```
%hist(vd,z)
VD=vd;
%-----
% Questo programma calcola la distanza tra due funzioni stimate f1 ed f2 %
%
function [MD]=epcq(R1, R2,a1,a2,b1,b2, trim, sd1, d1);

%funzioni differenze (distanza euclidea tra i parametri)

%d1R=d1/sd1;
d1R=d1;

%Calcolo della componente di adattamento relativa

d2=abs(R1-R2);
d3=abs(a1-a2);
d4=abs(b1-b2);

%Calcolo distanza globale combinazione delle due componenti
%trim=parametro d'importanza

MD=((trim*d1R)+(1-trim)*d2)+d3+d4;

% MD = Distanza epc
```





# Bibliography

- [1] Aitchison, J.(1986): The Statistical Analysis of Compositional Data, New York: Chapman Hall.
- [2] Benzecri, J.P. (1973): Theorie de l'information et classification d'aprues un tableau de contingence. L'Analyse des donnees, Tome 1, Dunod.
- [3] Bertrand P. et Goupil F. Descriptive statistics for symbolic data, In Symbolic official data analysis, Springer, 103-124, 1999.
- [4] Billard, L., Diday, E. (2003): From the Statistics of Data to the Statistics of Knowledge: Symbolic Data Analysis Journal of the American Statistical Association, 98, 462, 470-487.
- [5] Billard L. and Diday E. Regression analysis for interval-value data, In data analysis, classification and related methods, Eds. Kiers H., Rasson J., Groenen P. and Schader M., IFCS 2000.
- [6] Billard, L., Diday, E. (2006) Symbolic Data Analysis. Conceptual Statistics and Data Mining. Wiley Series in Computational Statistics.
- [7] Bock H-H. and Diday E. (eds.) Analysis of Symbolic Data. Exploratory methods for extracting statistical information from

- complex data. Springer Verlag, Heidelberg, 425 pages, ISBN 3-540-66619-2, 2000.
- [8] Brito P. Analyse de donnees symboliques: Pyramides d’heritage, Thèse de doctorat, Université Paris IX Dauphine, 1991.
- [9] Calinski, R.B. and Harabasz, J. (1974): A dendrite method for cluster analysis, *Communications in Statistics*, 3, 1-27.
- [10] Canal L. and Pereira M.(1998) Towards statistical indices for numeroid data, in: *Proceedings of the NTTTS’98 Seminar*, Sorrento Italy.
- [11] Cazes P., Chouakria A., Diday E. et Schektman Y. Extension de l’analyse en composante principales á des données de type intervalle, *Rev. Statistique Appliquée*, Vol. XLV Num. 3 pag. 5-24, Francia, 1997.
- [12] Celeux, G.; Diday, E.; Govaert, G.; Lechevallier, Y.; Ralambondrainy, H. (1989): *Classification Automatique des Données*. Bordas, Paris
- [13] Chavent and Lechevallier (2002): Dynamical clustering algorithm of interval data: optimization of an adequacy criterion based on Hausdorff distance. Sokolowsky and Bock (eds): *Classification, Clustering and Data Analysis*, Springer, 53-59
- [14] Chavent, M., De Carvalho, F.A.T., Lechevallier, Y., and Verde, R. (2003): Trois nouvelles methodes de classifcation automatique des donnees symbolique de type intervalle, *Revue de Statistique Appliquee*, LI, 4, 5-29.
- [15] Chavent, M., De Carvalho, F.A.T., Lechevallier, Y., and Verde, R. (2006): new clustering methods for interval data, *Computational statistics*,Phisica,Verlag, 21, 211-229.

- [16] Chouakria A. Extension des méthodes d'analyse factorielle á des données de type intervalle, Thèse de doctorat, Université Paris IX Dauphine, 1998.
- [17] Csiszar, I. (1967): Information type measures of difference of probability distributions and indirect observations, *Studia Sci. Math. Hungar*, 2, 299-318.
- [18] De Boor Carl, A Practical Guide to Splines, Springer-Verlag, New York, 1978.
- [19] De Carvalho (2007): Fuzzy c-means clustering methods for symbolic interval data. *Pattern Recognition Letters*
- [20] De Carvalho, Brito and Bock (2006): Dynamic clustering of interval data based on L2 distance. *Computational Statistics*, 21 (2), 231-250.
- [21] De Carvalho, Souza, Chavent, Lechevallier (2006): Adaptive Hausdorff distances and dynamic clustering of symbolic interval data. *Pattern Recognition Letters*, 27, 167-179
- [22] Di Zaven A. Karian, Edward J. Dudewicz *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*, CRC PRESS.
- [23] Diaconis, P. (1988). *Group Representations in Probability and Statistics*, Institute of Mathematical Statistics, Harvard University, CA.
- [24] Diday E. Introduction l'approche symbolique en Analyse des Données. Première Journées Symbolique-Numérique. Université Paris IX Dauphine. Décembre 1987.

- [25] Diday E. L'Analyse des Données Symboliques: un cadre théorique et des outils. Cahiers du CEREMADE, 1998.
- [26] Diday, E. (1971): Le methode des nuees dynamique, Revue de Statistique Appliquee, 19, 2, 19-34.
- [27] Diday, E. and Govaert, G. (1977) : Classification Automatique avec Distances Adaptatives. R.A.I.R.O. Informatique Computer Science,11 (4) 329–349
- [28] Diday, E., and Simon, J.C. (1976): Clustering analysis, In: Fu, K.S. (Eds.), Digital Pattern Recognition, 47-94, Springer Verlag, Heidelberg.
- [29] Escofier, B. Pagés, J. (1988-1998), Analyses factorielles simples et multiples; objectifs, méthodes et interprétation, Dunod.
- [30] Escofier, B. Pagés, J. (1994), Multiple factor analysis (afmult package), Computational statistics & data analysis 18, 121-140.
- [31] Ferson Scott, Kreinovich Vladik, Hajagos Janos, Oberkampf William and Ginzburg Lev SAND2007 “Experimental uncertainty estimation and statistics for data having interval uncertainty” Applied Biomathematics 100 North Country Road Setauket, New York.
- [32] Gibbs, A.L. and Su, F.E. (2002): On choosing and bounding probability metrics, International Statistical Review, 70, 419.
- [33] Gioia F.(2001) Statistical Methods for Interval Variables, Ph.D. thesis, Dep. of Mathematics and Statistics -University Federico II Naples, in Italian.
- [34] Gioia, F. and Lauro, N.C. (2006) Principal Component Analysis on Interval Data, Computational statistics, In press.

- [35] Hellinger, E. (1907): Die Orthogonalinvarianten quadratischer Formen von unendlich vielen Variablen, Dissertation, GÄotttingen.
- [36] Hickey T., Ju Q. and Van Emden M.H.(2001) Interval arithmetic: From principles to implementation, Journal of the ACM, 48, 5, 1038-1068.
- [37] Huber, P.J. (1981): Robust statistic, John Wiley and Sons, New York.
- [38] Irpino A., Lauro C.N. and Verde R.(2003) Visualizing symbolic data by closed shapes, in: Between Data Science and Applied Data Analysis, Schader M., et al. eds., GfKl, Springer Verlag, Heidelberg, Studies in Classification, Data Analysis, and Knowledge Organization., 44-251.
- [39] Irpino A., Verde R., and Lechevallier Y. (2006): Dynamic clustering of histograms using Wasserstein metric, in COMPSTAT 2006, (Eds. Rizzi, Vichi), Springer, Berlin,869-876.
- [40] Irpino, A. and Verde, R.(2005): A New Distance for Symbolic Data Clustering, CLADAG 2005, Book of short papers, MUP, 393-396.
- [41] Lauro C. and Palumbo F.(1998) New approaches to principal components analysis on interval data, in: NTTS'98,Sorrento, Italy, vol. 2.
- [42] Lauro C.N. and Palumbo F.(2000) Principal component analysis of interval data: A symbolic data analysis approach, Computational Statistics, 15, 1, 73-87.
- [43] Lauro C.N., Palumbo F. and Iodice D'Enza A.(2003) New graphical symbolic objects representations in parallel coordinates, in:

- Between Data Science and Applied Data Analysis, Schader M. et al. eds., GfKl, Springer Verlag, Heidelberg, Studies in Classification, Data Analysis, and Knowledge Organization, 288-295.
- [44] Lauro C.N., Verde R. and Palumbo F.(2000) Factorial methods with cohesion constraints in symbolic objects, in: IFCS'00.
- [45] Mallows, C. L. (1972): A note on asymptotic joint normality. *Annals of Mathematical Statistics*, 43(2),508-515.
- [46] Lawson, C. L. and Hanson, R. J. (1974) *Solving Least Squares Problems*. Prentice Hall.
- [47] Moore R.E.(1966) *Interval Analysis*, Prentice Hall, Englewood Cliffs, NJ.
- [48] Palumbo F. and Lauro C.N.(2003) A PCA for interval valued data based on midpoints and radii, in: *New developments in Psychometrics*,
- [49] Yanai H. et al. eds., *Psychometric Society*, Springer-Verlag, Tokyo.
- [50] Rodríguez O. *Classification et Modèles Linéaires en Analyse des Données Symboliques*, Thèse de doctorat, Université Paris IX Dauphine, 2000.
- [51] Rodríguez O., Diday E., Winsberg S., (2000) Generalization of the Principal Components Analysis to Histogram Data presented to the 4th European Conference on Principles and Practice of Knowledge Discovery in Data Bases, Lyon, France.
- [52] Romano E., Giordano G., Lauro N.C. (2005) An inter-models distance for clustering utility functions, presented to the 3rd world

- conference on Computational Statistics & Data Analysis, Limassol, Cipro, submitted to *Statistica Applicata*.
- [53] Souza and De Carvalho (2004): Clustering of interval data based on city-block distances. *Pattern Recognition Letters*, 25 (3), 353-365
- [54] Tran and Duckstein (2002): Comparison of fuzzy numbers using a fuzzy distance measure, *Fuzzy Sets and Systems*, 130, 331-341
- [55] Verde R., Irpino A. (2006). A new Wasserstein based distance for the hierarchical clustering of histogram symbolic data. *Data Science and Classification* (Eds. Batanjeli, Bock, Ferligoj, Ziberna), Springer, Berlin, pp. 185-192.
- [56] Ward, J.H. (1963): Hierarchical Grouping to Optimize an Objective Function, *Journal of the American Statistical Association*, vol. 58, 238-244.